UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Rafael da Rosa Righi

P2P-Role: Uma Arquitetura de Controle de Acesso Baseada em Papéis para Sistemas Colaborativos Peer-to-Peer

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação.

Profa. Carla Merkle Westphall, Dra.
Orientadora

Florianópolis, Abril de 2005

P2P-Role: Uma Arquitetura de Controle de Acesso Baseada em Papéis para Sistemas Colaborativos Peer-to-Peer

Rafael da Rosa Righi

Esta Dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Raul Sidnei Wazlawick, Dr.
Coordenador do Curso

Banca Examinadora

Profa. Carla Merkle Westphall, Dra.
Orientadora

Prof. Antônio Augusto Medeiros Fröhlich, Dr.

Prof. Elias Procópio Duarte Júnior, Dr

Prof. Vitório Bruno Mazzola, Dr.

“A adversidade faz do homem um sábio.” (Sêneca, filósofo espanhol)

A minha família e amigos.

Agradecimentos

Agradeço a Deus pelas oportunidades que ele coloca na minha caminhada de vida e por chegar ao fim de mais uma etapa: o mestrado acadêmico.

Agradeço a minha família pela força que tenho recebido. Agradeço ao meu irmão Rodrigo pelos conselhos, conversas e apoio dados durante o mestrado. Em especial, agradeço a minha esposa Luciana que soube compreender as horas que fiquei longe me dedicando à pesquisa da arquitetura P2P-Role.

Agradeço a minha orientadora Carla Westphall pelos ensinamentos passados e pelo tempo desprendido em reuniões realizadas durante o desenvolvimento deste trabalho. As revisões dos artigos publicados e as sugestões fornecidas para a dissertação foram fundamentais para a realização de um mestrado com qualidade.

Agradeço aos meus colegas e amigos do Laboratório de Redes e Gerência (LRG-UFSC) pelo auxílio e apoio concedidos. E também aos meus companheiros do Ponto de Presença da RNP em Santa Catarina (PoP-SC) pela amizade e parceria. Em especial, agradeço ao Guilherme Rhoden e ao Edison Melo pela força dada durante o período que fui bolsista do PoP-SC e pela parceria no projeto de Gerência de Tráfego Peer-to-Peer na RNP.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ACL	*Access Control Lists*
API	*Application Programming Interface*
CA	*Certificate Authority*
CPU	*Central Processing Unit*
DAC	*Discretionary Access Control*
DHT	*Distributed Hash Table*
DNS	*Domain Name Server*
DoD	*Department of Defense*
DSD	*Dynamic Separation of Duties*
ERP	*Endpoint Routing Protocol*
HTTPS	*HyperText Transfer Protocol, Secure*
IP	*Internet Protocol*
IRC	*Internet Relay Chat*
JDBC	*Java Database Connectivity*
JXTA	*Juxtapose*
MAC	*Mandatory Access Control*
NAT	*Network Address Translation*
NIST	*National Institute of Standards and Technology*
PKI	*Public Key Infrastructure*
P2P	*Peer-to-Peer*
PDP	*Peer Discovery Protocol*
PIP	*Peer Information Protocol*

PBP	*Peer Binding Protocol*
PRP	*Peer Resolver Protocol*
RBAC	*Role-Based Access Control*
RVP	*Rendezvous Protocol*
SGBD	Sistema de Gerência de Banco de Dados
SOAP	*Simple Object Access Protocol*
SoD	*Separation of Duties*
SSD	*Static Separation of Duties*
SSL	*Secure Socket Layer*
TCP	*Transfer Control Protocol*
TCSEC	*Trusted Computer Security Evaluation Criteria*
THA	*Trust-Holding Agent peers*
TLS	*Transport Layer Security*
TTL	*Time to Live*
UUID	*Universal Unique Identifier*
W3C	*World Wide Web Consortium*
WWW	*World Wide Web*
XML	*eXtensible Markup Language*

Resumo

Os sistemas Peer-to-Peer apresentam uma forma de computação distribuída onde cada participante atua como cliente e servidor de recursos. Entre os principais desafios existentes nesse tipo de computação estão o desenvolvimento de técnicas para incentivar a colaboração entre os usuários e a proteção dos elementos e informações localizados no ambiente Peer-to-Peer. Este documento define uma arquitetura de controle de acesso baseada em papéis específica para as redes Peer-to-Peer e busca, assim, contribuir para a construção de sistemas colaborativos mais robustos e seguros. Também são exploradas as relações entre o controle de acesso e as técnicas utilizadas para estimular a cooperação entre os integrantes da rede Peer-to-Peer. Pretende-se, desta forma, que a arquitetura de controle de acesso definida também auxilie para minimizar a quantidade de nós que não colaboram com a rede e que apenas sugam seus recursos. O protótipo implementado baseia-se nas tecnologias JXTA e P2PSockets e provê meios para que cada nó da rede P2P gerencie a política de segurança dos seus recursos.

Abstract

Peer-to-Peer networks are distributed systems where each user acts at the same time as a client and server of resources. Among the main existing challenges of this type of computation are the development of techniques to stimulate the contribution between the users and the protection of the members and informations. This document defines a role-based access control architecture specific for Peer-to-Peer networks and, in this way, contributes for the construction of collaborative systems that are more robust and safe. The relations between the access control and the techniques to minimize the number of free riders that do not assist the net and just suck its resources are also explored. The developed architecture intends to contribute for the development of Peer-to-Peer applications less vulnerable to the actions of free riders. The prototype implemented is based on the JXTA and P2PSockets technologies and provides means for security policy management of resources in each node of Peer-to-Peer network.

Capítulo 1

Introdução

As redes Peer-to-Peer são sistemas distribuídos sem controle centralizado, nos quais o programa que é executado em cada integrante da rede é equivalente em funcionalidade. Os participantes do sistema Peer-to-Peer podem agir como clientes ou servidores de recursos. Eles constroem uma rede virtual sob a rede física (geralmente a Internet) e colaboram uns com os outros em tarefas específicas, como o compartilhamento de arquivos. Esse tipo de computação explora as bordas da rede, representa uma alternativa ao modelo cliente-servidor largamente conhecido e possui um potencial ainda não explorado em sua totalidade [LOO 03].

O impulso inicial para os estudos em redes Peer-to-Peer proveio do sucesso de aplicações P2P como o Naspter e o Gnutella e do impacto (revolução) causado por elas a partir de 2000 [ROU 04]. No princípio, o enfoque das pesquisas científicas concentrava-se em desenvolver ambientes P2P estáveis, que tivessem boa escalabilidade e mecanismos de busca robustos. O crescimento e o amadurecimento das aplicações P2P e o conseqüente avanço das pesquisas possibilitaram o estudo e a investigação de outros requisitos apresentados pelas redes Peer-to-Peer como a inclusão de mecanismos de reputação no ambiente colaborativo e a construção de aplicações mais seguras e confiáveis.

O foco da maioria das aplicações Peer-to-Peer é o compartilhamento de arquivos. Esse gênero de aplicações foi o responsável por popularizar o termo Peer-to-Peer e o seu uso entre os usuários finais [SEN 04]. Contudo, as possibilidades de utilização do para-

digma Peer-to-Peer vão muito além do compartilhamento de arquivos. Outros cenários onde se pode utilizar a computação aos pares incluem o comércio eletrônico entre usuários de uma comunidade virtual, a construção de ambientes que permitam o trabalho colaborativo (por exemplo, a edição simultânea de documentos) [EIK 04] e a exploração do poder de processamento ocioso presente nas bordas da rede [KOR 01].

Cada tipo de aplicação Peer-to-Peer possui seus próprios requisitos de segurança. Aplicações Peer-to-Peer voltadas ao comércio eletrônico entre organizações exigem maior comprometimento com a proteção se comparado aos programas para o compartilhamento de músicas na Internet. Entre os requisitos fundamentais de segurança está o controle de acesso [VLA 04, DAS 03]. No contexto de uma rede Peer-to-Peer, o controle de acesso permite que cada nó ou comunidade coloque em prática sua própria política de autorização. Assim, cada participante pode estabelecer quem são as entidades que podem acessar os seus recursos e quais as condições que elas devem cumprir para terem esse direito.

Além dos requisitos de segurança, o êxito de uma rede Peer-to-Peer depende de fatores como o protocolo de comunicação utilizado, a possibilidade de operar em ambientes protegidos (por exemplo, aqueles que adotam *firewalls*), a arquitetura de distribuição empregada (totalmente descentralizada ou não), o tempo decorrido desde uma solicitação até o recebimento de respostas e o número de usuários participantes do sistema. Essa última característica é especialmente importante, pois é ela quem determina o tamanho da comunidade virtual e o volume de recursos presente na rede Peer-to-Peer.

O tema central deste documento é o controle de acesso para as redes P2P. A adição de uma arquitetura de controle de acesso a um ambiente Peer-to-Peer agrega qualidade e proteção ao sistema colaborativo. A seção 1.1 apresenta os objetivos desta dissertação de mestrado. A seção 1.2 exibe as motivações e justificativas que impulsionaram o desenvolvimento da pesquisa na área de controle de acesso e autorização para sistemas Peer-to-Peer. Por fim, mostra-se na seção 1.3 a estrutura da dissertação e a ordem dos capítulos que a compõem.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral desta dissertação é definir e desenvolver uma arquitetura de controle de acesso para redes Peer-to-Peer denominada P2P-Role que contenha as seguintes características: *(i)* facilidade de uso e configuração pelos usuários finais; *(ii)* seja estruturada sob uma plataforma de desenvolvimento de aplicações P2P conhecida; *(iii)* possa ser utilizada por outras aplicações P2P que requerem controle de acesso sob os recursos localizados na rede virtual; *(iv)* utilize o modelo de controle de acesso baseado em papéis (RBAC) como centro do processo de autorização; *(v)* efetivamente contribua para a existência de redes Peer-to-Peer mais seguras e robustas. A arquitetura P2P-Role, além de ter bons fundamentos conceituais, deve ser aplicável em ambientes reais e suprir as principais deficiências apresentadas pelos trabalhos relacionados.

1.1.2 Objetivos Específicos

Os objetivos específicos são:

- adicionar ao P2P-Role técnicas que incentivam a colaboração entre os usuários da rede Peer-to-Peer. A arquitetura resultante desta união deve, além de proporcionar meios para efetivar o controle de acesso no sistema, auxiliar para a diminuição dos nós parasitas da rede (aqueles que apenas sugam os recursos do sistema) e para a construção de um ambiente Peer-to-Peer mais justo;

- utilizar os projetos JXTA [HAL 03] e P2PSockets [NEU 04] no processo de construção do protótipo. Esses projetos são reconhecidos pela comunidade acadêmica e representam o estado da arte na área de desenvolvimento de aplicações Peer-to-Peer;

- desenvolver uma aplicação gráfica que possibilite cada participante da rede Peer-to-Peer visualizar na forma de grafos a organização de seu modelo de controle de acesso RBAC.

1.2 Motivações e Justificativas

Entre as motivações que estimularam este trabalho está a grande evolução e importância adquiridas pelas redes Peer-to-Peer e a pesquisa elaborada por Daswani e Garcia-Molina [DAS 03] (*Stanford Peers Group*), que relaciona os problemas de segurança ainda não solucionados nas redes Peer-to-Peer. Estes pesquisadores classificaram as necessidades de segurança das comunidades P2P em quatro grupos – disponibilidade, autenticidade de conteúdo, anonimidade e controle de acesso – e apresentaram o que falta em cada um deles. Em especial, o presente trabalho preocupa-se com o último grupo de requisitos de segurança, sobretudo com a construção de uma arquitetura de controle de acesso que propicie a cada usuário da rede estabelecer políticas de autorização para seus recursos.

O processo de controle de acesso é um dos pilares da segurança. Com relação a proteção de sistemas colaborativos, observa-se que originalmente o foco das aplicações P2P era criar um ambiente com uma funcionalidade específica e uma topologia que fosse estável para suportar a comunicação entre os elementos da rede. A computação Peer-to-Peer evoluiu e as necessidades de segurança também. As pesquisas em segurança de redes P2P se distribuem em vários temas, sendo os principais a criptografia e a infraestrutura de chaves públicas no contexto de sistemas colaborativos P2P. Entre as pesquisas mais atuais nessas áreas está a de Berket e Muratas [BER 04] (*Berkeley National Laboratory*).

Apesar de existirem inúmeros trabalhos científicos que abordam a identificação e autenticação em redes Peer-to-Peer, são escassos os que priorizam o controle de acesso nesses ambientes [BER 04]. Essa situação é comprovada, mesmo havendo estudos como [DAS 03] e [FEN 02] que relacionam o controle de acesso como questão fundamental para a expansão do paradigma Peer-to-Peer para outros domínios, como os leilões distribuídos. A pesquisa descrita nesta dissertação concentra seu foco no controle de acesso para redes P2P e procura, assim, preencher algumas lacunas ainda não exploradas e avançar com o estado da arte na sua área de atuação.

Como exposto na seção de objetivos específicos, um dos propósitos desta dissertação é agregar à arquitetura P2P-Role a capacidade de distinguir e controlar os participantes da rede Peer-to-Peer que agem como parasitas de outros nós. Busca-se, com esta

união, formar um sistema que regule o comportamento dos nós na rede Peer-to-Peer. A justificativa central para unir estas duas áreas de pesquisa (controle de acesso e incentivo a colaboração) é possibilitar a existência de redes que sejam, além de seguras, mais justas e harmoniosas.

As técnicas que incentivam a colaboração nas redes Peer-to-Peer procuram diminuir os efeitos de um episódio chamado “tragédia dos comuns” [HAR 68]. Esta lei afirma que os usuários sugarão ao máximo os recursos da rede se estes puderem ser acessados sem restrições – tendência ao benefício próprio em detrimento do coletivo. A existência de nós caronas auxilia para aumentar o congestionamento dos enlaces e concentrar as conexões sob aqueles nós que dispõem recursos no sistema. As pesquisas na área de incentivo a colaboração em redes P2P emergiram em 2000 e o contínuo estudo desse tema ([STR 04, PAP 04]) reforça sua relevância e o quanto é vital para a existência de ambientes colaborativos de qualidade.

1.3 Estrutura do Documento

Este documento é composto por oito capítulos. A seguir é descrita a ordem de apresentação dos capítulos e o tema tratado por cada um deles.

O capítulo 2 apresenta os principais conceitos relacionados à computação Peer-to-Peer. Ele descreve quais são as peculiaridades e características das redes P2P, como elas são classificadas, quais as decisões de projeto que precisam ser consideradas durante o processo de desenvolvimento de um sistema P2P e o funcionamento básico desse tipo de aplicação. O objetivo desse capítulo é transmitir informações sobre as redes Peer-to-Peer, de forma que seja possível compreender com clareza os capítulos seguintes deste documento.

O capítulo 3 descreve os projetos JXTA e P2PSockets. São exibidos seus objetivos, benefícios e funcionamento. Estes projetos foram utilizados no protótipo desenvolvido e sua compreensão é importante para conhecê-lo em sua totalidade.

O capítulo 4, em um primeiro momento, expõe a definição de segurança computacional e faz uma relação entre as propriedades, mecanismos e políticas de segurança.

Logo após, ele direciona seu foco para a autorização e para os modelos de controle de acesso. São mencionados os modelos de controle de acesso discricionário (DAC), obrigatório (MAC) e o baseado em papéis (RBAC). Também são citadas as principais vantagens do modelo RBAC sobre os demais.

Os próximos capítulos apresentam a contribuição efetiva deste trabalho. O capítulo 5 faz uma ampla revisão bibliográfica a respeito da segurança para redes Peer-to-Peer. São explorados temas como reputação, micropagamentos, utilização de infraestrutura de chaves públicas em redes Peer-to-Peer, técnicas que incentivam a colaboração na rede, entre outros. Ele também discute o controle de acesso para ambientes Peer-to-Peer. É nesse capítulo que são apresentados os principais trabalhos relacionados ao assunto tratado neste documento.

O capítulo 6 descreve a arquitetura de controle de acesso P2P-Role, além de seus componentes, vantagens e funcionamento. Insere-se ainda neste capítulo como as técnicas que incentivam a colaboração nas redes P2P são agregadas ao P2P-Role e quais os benefícios desta união. Esses assuntos concentram o foco principal desta dissertação de mestrado.

O protótipo elaborado no decorrer da pesquisa e os experimentos realizados estão descritos no capítulo 7. São expostos detalhes sobre a implementação do protótipo, as decisões de projeto executadas, o modelo RBAC desenvolvido em cada nó da rede Peer-to-Peer e como as tecnologias JXTA e P2PSockets foram aproveitadas no sistema P2P construído.

O trabalho encerra no capítulo 8 com a conclusão, a qual retoma os principais resultados e contribuições desta dissertação de mestrado. Também são apresentados possíveis complementos sobre a pesquisa, a título de trabalhos futuros.

Por fim, ressalta-se que cada capítulo contém uma seção chamada Balanço. Ela é a última seção de cada capítulo e sua função é reunir e compilar os principais aspectos apresentados em determinada parte do documento.

Capítulo 2

Redes Colaborativas Peer-to-Peer

As redes Peer-to-Peer são sistemas distribuídos sem controle centralizado ou organização hierárquica, nas quais o programa que é executado em cada elemento é equivalente em funcionalidade [CAV 04]. Esses sistemas possibilitam que os usuários sejam, além de consumidores de recursos, os próprios responsáveis por disponibilizá-los. Por minimizar o papel dos elementos centralizadores, os sistemas P2P tendem a ser imunes à censura, monopólios, regulamentos e outros exercícios atribuídos às autoridades centralizadoras [AGR 03].

As aplicações para a distribuição de arquivos foram as responsáveis pela popularização das redes colaborativas P2P. Os equipamentos conectados no ambiente colaborativo formam uma rede virtual sobre a rede de dados subjacente (IP, no caso da Internet). Sistemas P2P trazem a conectividade para as bordas da rede, permitindo que qualquer equipamento conectado se comunique e colabore com os demais [SAD 03].

A definição de sistemas Peer-to-Peer não é consenso na comunidade científica, principalmente devido às semelhanças entre computação colaborativa P2P, computação em grade (*grid*) e aglomerado de computadores (*cluster*) [TAL 03]. Os sistemas são classificados como Peer-to-Peer quando compreendem o conjunto de características inerente a esse tipo de computação colaborativa.

O tema Peer-to-Peer está organizado em 6 partes. A seção 2.1 descreve as particularidades que os sistemas Peer-to-Peer apresentam. A classificação dos sistemas Peer-

to-Peer está presente na seção 2.2. As questões de projeto que precisam ser examinadas durante o processo de construção de um sistema P2P são o tema da seção 2.3. A seção 2.4 exibe duas aplicações Peer-to-Peer conhecidas e discute seus modos de funcionamento. A seção 2.5 apresenta as pesquisas que abordam a gerência e a monitoração do tráfego Peer-to-Peer. O capítulo 2 termina na seção 2.6 com a seção de Balanço, a qual retoma os principais tópicos mencionados no corrente capítulo.

2.1 Características do Sistemas Peer-to-Peer

Os sistemas Peer-to-Peer possibilitam o compartilhamento de recursos e a cooperação entre os integrantes de uma comunidade virtual. Para alcançar esse objetivo os sistemas P2P empregam o conjunto de características presentes na Figura 2.1. Esta seção preocupa-se em exibí-las, de modo que se compreenda como cada característica auxilia para a construção dos modelos Peer-to-Peer.

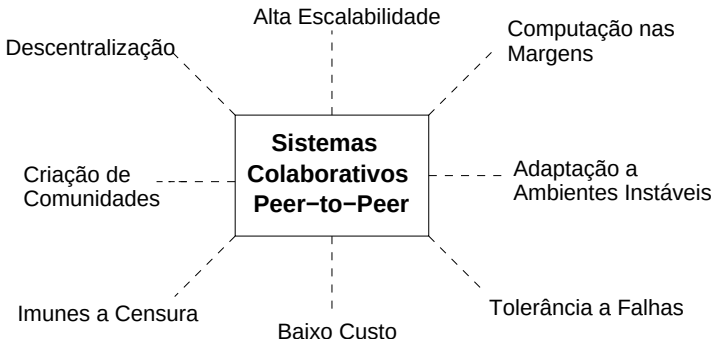

Figura 2.1: Características das redes Peer-to-Peer

O modelo de computação Peer-to-Peer possui várias diferenças em relação ao esquema cliente-servidor, amplamente implementado na Internet. No padrão de comunicação cliente-servidor o fluxo de transmissão é assimétrico, pois a maioria do tráfego ocorre no sentido servidor→cliente, sendo o cliente, na maioria da vezes, um participante passivo na transmissão. Nos sistemas P2P todos os integrantes da rede possuem o mesmo programa (código-fonte) e cada membro da rede pode atuar como cliente e servidor de

recursos.

Essa característica torna a transmissão no ambiente Peer-to-Peer mais simétrica, pois o nó P2P pode abrir conexões para outros pontos da rede e também receber e tratar conexões dos demais participantes do ambiente colaborativo. A comunicação simétrica entre os dispositivos de uma rede não é um paradigma criado pela computação Peer-to-Peer. O projeto inicial da Internet previa um modelo sem a existência de servidores centralizados, onde todos os membros da infraestrutura eram equiparáveis em funcionalidade [LOO 03].

Conforme relata Sadok et al [SAD 04], aos poucos a Internet foi perdendo essa característica, por motivos como a especialização de funções, necessidade de robustez e alta disponibilidade, problemas de segurança e limitação de recursos computacionais ou capacidade dos enlaces de comunicação. Nesse momento, o modelo predominante é aquele onde alguns poucos servidores superdimensionados prestam serviços a uma infinidade de clientes nas bordas da rede. Isso ocorre mesmo que a grande maioria dos clientes tenha alta capacidade de processamento e armazenamento. A Figura 2.2 exibe uma comparação entre o modelo cliente-servidor e o definido pela computação Peer-to-Peer.

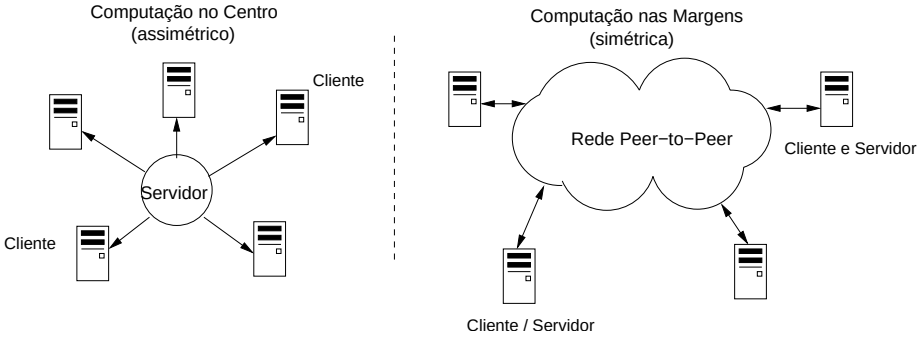

Figura 2.2: Comparação entre computação no centro e nas margens

A descentralização é outra propriedade das redes Peer-to-Peer. Ao contrário do modelo cliente-servidor que dispõe os recursos no servidor, o ambiente colaborativo distribui o seu volume de recursos entre os envolvidos na rede. Percebe-se, portanto, que as redes P2P necessitam mecanismos de busca de informações especializados. A forma como um sistema Peer-to-Peer procede para encontrar um recurso na rede virtual deter-

mina a sua classificação em “puro” (processo totalmente descentralizado) ou “híbrido”. A seção 2.2 expõe a classificação das redes Peer-to-Peer e apresenta exemplos que compõem cada um dos tipos mencionados.

A natureza descentralizada do modelo Peer-to-Peer é a origem de outras características desses sistemas. A disponibilidade dos recursos melhora com a descentralização, já que um número maior de informações está disponível para os usuários legítimos. Outra peculiaridade dos sistemas P2P realçada com a distribuição dos recursos é a tolerância a falhas. Caso um elemento deixe subitamente o sistema, a rede Peer-to-Peer adapta-se ao seu novo estado e continua a prover sua funcionalidade normalmente. Essa característica também dificulta os ataques de negação de serviço realizados contra o sistema Peer-to-Peer, pois para paralisar o sistema é necessário que uma grande parcela de seus elementos estejam desativados.

Uma rede Peer-to-Peer pode ser considerada uma rede *overlay*[1]. As redes *overlay* são redes virtuais criadas sobre uma rede já existente. Elas constroem uma arquitetura de nível mais alto de abstração capaz de esconder alguns detalhes de comunicação atribuídos as camadas inferiores de uma infraestrutura de rede [AND 01]. A Figura 2.3 mostra uma rede *overlay* e sua respectiva rede física.

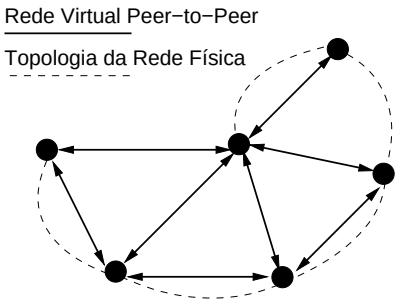

Figura 2.3: Rede *overlay* [MIL 02]

Entre as propriedades que auxiliaram para a difusão dos sistemas colaborativos Peer-to-Peer está o suporte à operação em ambientes instáveis. As aplicações P2P se adequam a topologias que empregam *firewalls* com regras rígidas, endereços IP dinâmicos

[1]O termo *overlay*, quando traduzido para a língua portuguesa, é comumente chamado de revestimento ou cobertura.

nos dispositivos de rede e a ambientes que utilizam a tradução de endereços de rede (NAT) para possibilitar a comunicação entre elementos com endereços IP privados e redes localizadas na Internet. A seção 2.3 expõe em detalhes os obstáculos que os sistemas Peer-to-Peer devem superar para garantir a comunicação entre todos os elementos da rede e aponta as soluções adotadas pelas aplicações P2P para funcionarem em ambientes instáveis.

A utilização dos sistemas Peer-to-Peer propicia a criação de comunidades de usuários. Uma comunidade de usuários possui interesses comuns e, como toda sociedade, deve seguir regras de comportamento que regulam o bom andamento das interações na rede, garantindo, assim, o convívio harmonioso de seus membros. O número de elementos que compõem uma rede Peer-to-Peer é aleatório e a topologia de comunicação existente é variável e constantemente modificada. Os nós agregam-se à comunidade P2P utilizando outros nós como porta de entrada e não há nenhuma restrição quanto ao tempo mínimo que um elemento deve permanecer na rede virtual estabelecida. Os membros da rede podem desligar-se da comunidade Peer-to-Peer sem prévio aviso e a qualquer momento; questão que reforça a característica de instabilidade das redes P2P.

2.2 Classificação das Redes Peer-to-Peer

Existem diversas classificações para as redes Peer-to-Peer. Entre os fatores determinantes nessas organizações estão o tipo de procura implementado na rede Peer-to-Peer e a existência (ou inexistência) de nós com diferentes funções no sistema P2P.

A classificação para as redes Peer-to-Peer apresentada no estudo "P2P *Architect Project*" [P2P 04], de forma geral, separa os modelos P2P nos grupos descentralizado e semicentralizado. Nas redes Peer-to-Peer totalmente descentralizadas não há nenhuma entidade central responsável por intermediar o processo de busca e divulgação de recursos. Já o modelo semicentralizado prevê um nó central para gerir as informações de controle ou um conjunto de super-nós que assume tais funções (onde a queda de um super-nó afeta apenas os nós inferiores ligados a ele [SAD 04]).

Segundo Schollmeier [SCH 01], as redes Peer-to-Peer podem ser divididas em puras e híbridas. Ele afirma que uma arquitetura de rede Peer-to-Peer é considerada pura se

ela não utiliza nenhum elemento centralizado para prover as funções básicas de uma rede P2P (por exemplo, busca por recursos) e, principalmente, se qualquer um dos elementos do sistema possa ser retirado da topologia sem que a mesma sofra algum dano ou pare de fornecer seus serviços (relaciona a igualdade de função dos membros da rede). Nesse contexto, uma rede Peer-to-Peer híbrida é aquela que utiliza um ponto centralizado (ou vários) para prover alguns dos serviços necessários à rede Peer-to-Peer. A Figura 2.4 expõe a classificação das redes Peer-to-Peer conforme os trabalhos [SCH 01] e [P2P 04].

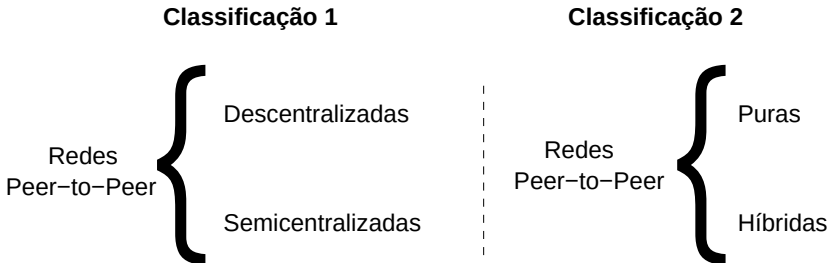

Figura 2.4: Classificação das redes Peer-to-Peer

Em outra classificação [LV 02], observa-se a presença de três tipos de arquiteturas Peer-to-Peer: *i*) a Centralizada; *ii*) a Descentralizada e Estruturada; *iii*) a Descentralizada e Não-Estruturada. Essa organização é mais completa que as anteriores e também mais aceita junto à comunidade científica. A Tabela 2.1 exibe a classificação definida por Qin Lv [LV 02] e cita exemplos de sistemas Peer-to-Peer para cada um dos grupos relatados.

Classificação	**Exemplo de Aplicações Peer-to-Peer**
Centralizado	Napster, MSN *Messenger*, ICQ
Descentralizado e Estruturado	Sistemas científicos como Chord, Pastry, Tapestry, CAN
Descentralizado e Não Estruturado	Gnutella, KaZaA

Tabela 2.1: Arquiteturas de rede P2P [LV 02]

Entre os representantes da arquitetura centralizada estão o Napster (aplicação para o compartilhamento de músicas) e os sistemas de troca de mensagens, como o MSN

Messenger (possibilita a comunicação em tempo real entre usuários localizados normalmente nas bordas da rede). No Napster a função do ponto central é catalogar e indexar as informações localizadas nos membros da rede. Nesse esquema, o elemento entra em contato com o servidor central quando está a procura de uma informação. Logo após receber os resultados do servidor, o participante da rede decide qual resposta lhe parece ser a melhor e abre uma conexão diretamente com o provedor do recurso, estabelecendo a computação Peer-to-Peer. Nos sistemas de trocas de mensagens, a função principal do elemento central é armazenar informações sobre os usuários da rede e quais deles estão ativos (*on-line*) em determinado instante. Um usuário, nesse sistema, utiliza o servidor central para conhecer quais dos seus contatos estão ativos no momento e, logo após, abre uma comunicação direta com algum deles.

A utilização de um meio centralizador pode ser positiva, pois diminui a complexidade da computação nos pontos. Porém, ele será um elemento central de falha – caso ele fique indisponível a rede fica sem funcionalidade. A Figura 2.5 apresenta as etapas existentes em uma comunicação na rede Napster (essa configuração encaixa-se também no modelo híbrido classificado por Schollmeier [SCH 01]). O primeiro passo que os membros da rede executam é a divulgação dos seus recursos a uma estação central. Essa estação concentra a identificação do volume de recursos existente no ambiente Peer-to-Peer. Os nós pesquisam a estação central para encontrar as informações que procuram. Conseqüentemente, a visão da rede Peer-to-Peer é a mesma para todos os seus integrantes.

As redes Peer-to-Peer consideradas “Descentralizadas e Estruturadas” possuem uma topologia controlada e os recursos do sistema são posicionados em locais que possibilitam a sua localização mais rapidamente [SAD 04]. Esses sistemas utilizam uma tabela de *hash* distribuída (DHT) [ZHU 05, BUE 03] para registrar os recursos e possibilitar uma visão uniforme da rede [BAL 03]. A utilização de uma DHT na rede Peer-to-Peer aumenta a eficiência da procura de uma informação (o tempo decorrido desde o lançamento da requisição até o recebimento de respostas é menor), porém agrega uma complexidade adicional aos elementos que compõem esse tipo de sistema. Outras vantagens desse modelo são o menor tráfego gerado a cada busca realizada e a facilidade de encontrar os recursos, mesmo aqueles raros. A seção 2.3 especifica o algoritmo de roteamento DHT e

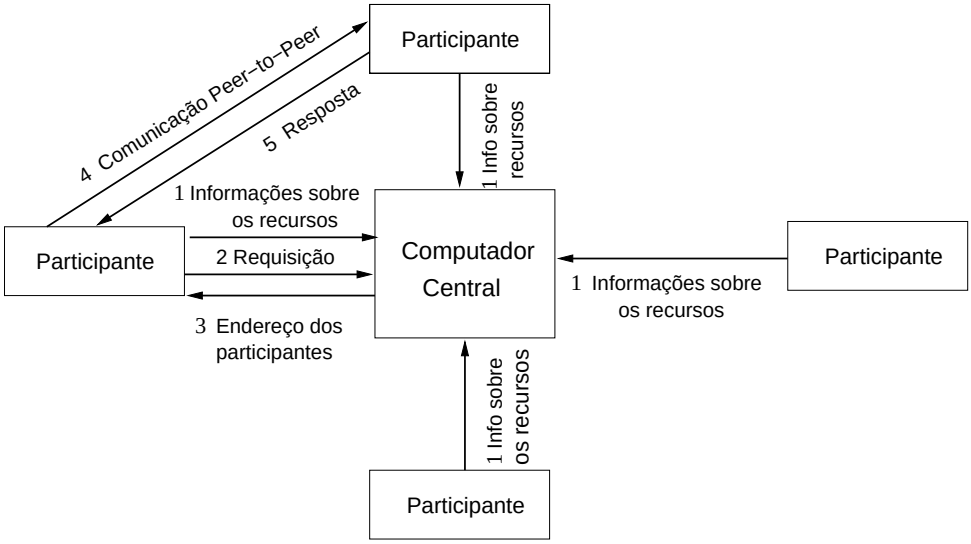

Figura 2.5: Troca de mensagens na rede Napster

mostra um exemplo de seu funcionamento.

Na classificação “Descentralizada e Não-Estruturada” não existe nenhum controle sobre a topologia da rede, nem um método de procura de recursos estruturado. Nessas redes nem sempre os membros da comunidade virtual conseguem visualizar a topologia da mesma forma, ou seja, dois nós que realizam a mesma procura podem receber respostas diferentes da rede. A seção 2.4 demonstra um ambiente onde esse fato ocorre.

A rede Peer-to-Peer Gnutella é um dos principais exemplos do modelo “Descentralizado e Não-Estruturado”. Seu modelo de procura por recursos é conhecido como busca por inundação[2]. Em geral, o protocolo utilizado nesse processo é o seguinte: quando um vizinho recebe um pedido ele verifica se não contém a informação requisitada; caso possuir, responde prontamente ao requisitor. Caso contrário, ele reenvia o pedido a seus vizinhos, executando o mesmo protocolo [BAL 03]. As mensagens de procura por recursos que os participantes da rede trocam possuem um contador de saltos e um número de identificação. Essas características informam que as mensagens de procura trafegam na rede até seu tempo de vida expirar e, caso a mesma mensagem seja transmitida a um nó mais de uma vez, ela será descartada no mesmo momento (evita o *loop* de requisições

[2]Também chamado de *flooding*.

na topologia).

2.3 Questões de Projeto em Redes Peer-to-Peer

Esta seção discute as principais decisões e questões de projeto que devem ser tomadas durante a construção de um sistema Peer-to-Peer. Barkai [BAR 01] divide essas decisões em cinco áreas:

- comunicação
- endereçamento e descoberta
- disponibilidade
- segurança
- gerência de recursos

Dentre as cinco áreas definidas, dá-se ênfase nesse documento para as áreas de comunicação, endereçamento e descoberta, segurança. Em especial, as questões de segurança nas redes Peer-to-Peer são tratadas separadamente no capítulo 5, principalmente por serem as decisões de projeto mais relevantes no contexto desse trabalho.

O sucesso das aplicações Peer-to-Peer depende da comunicação direta entre os elementos da rede. Essas aplicações executam normalmente sob a Internet, a qual emprega um vasto conjunto de técnicas e ferramentas que dificultam ou impossibilitam esse tipo de comunicação de acontecer. Existem duas questões importantes que os sistemas P2P devem tratar. A primeira é o emprego de *firewalls* nas redes de computadores. O *firewall* é uma barreira implantada no limite entre a rede que se deseja proteger e a rede pública. Ele atua principalmente como filtro de pacotes e é quem regula quais informações podem entrar ou sair da rede protegida.

O *firewall* limita o tráfego bidirecional entre os elementos localizados dentro e fora da rede protegida e reduz o número de portas abertas. Essa característica impossibilita a abertura de uma comunicação com origem na rede pública e destino na rede privada.

Apenas é permitido o tráfego de um IP externo para a rede interna se o elemento interno já tiver aberto uma comunicação com ele. O número de porta utilizado pela aplicação P2P nem sempre estará aberto no *firewall*. Esse é um dos pontos que precisam ser avaliados no processo de implementação de um sistema P2P.

Outro obstáculo que os sistemas P2P precisam sobrepor é a atribuição dinâmica de endereços IP aos computadores e o uso de endereços IP privados juntamente com métodos de tradução de endereços de rede (NAT). Por padrão, um computador com endereço IP privado não pode comunicar-se com computadores dispostos na Internet. O inverso também não é possível. O NAT, muitas vezes realizado no mesmo elemento que funciona como *firewall*, traduz endereços de rede inválidos para endereços válidos e possibilita a comunicação que primeiramente não podia acontecer. A Figura 2.6 apresenta as topologias de rede que utilizam o NAT.

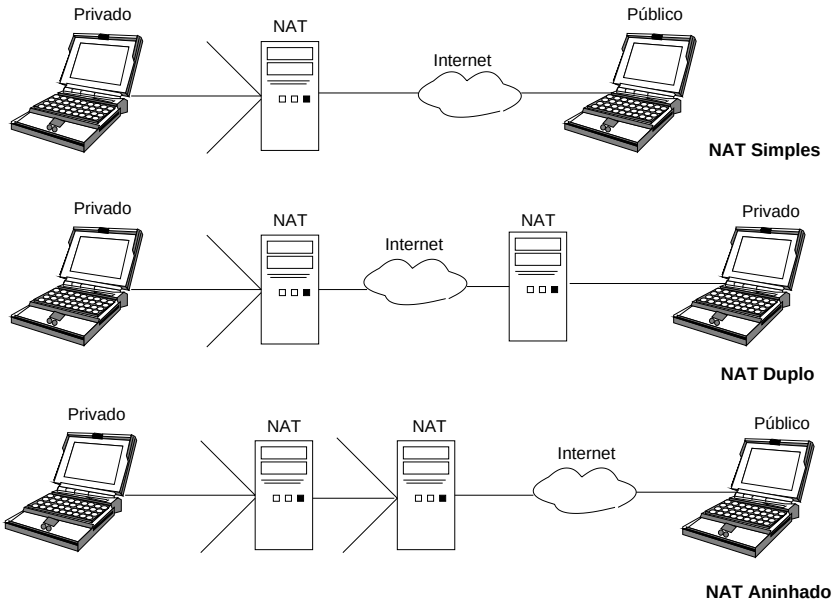

Figura 2.6: Topologias de rede com NAT [BAR 01]

Uma das soluções para o problema do controle de portas abertas no *firewall* é a utilização do protocolo HTTP e a porta 80. A porta 80 quase sempre é permitida na política de segurança adotada pela organização, pois o serviço de acesso a páginas hipertexto é um dos mais comuns na Internet. As aplicações podem encapsular suas mensa-

gens de controle e os próprios recursos através dos protocolos SOAP [FAR 03] e XML [CHA 04][3] e enviá-los utilizando o HTTP. Dá-se o nome de *http tunneling* a esse acontecimento. Além disso, algumas aplicações P2P possuem a capacidade de descobrir quais as portas permitidas no *firewall*. Elas usam essa informação para transpor a proteção desenvolvida na organização e manter as comunicações P2P ativas.

A solução para a inexistência de conexões bidirecionais em determinados *firewalls* é o emprego de um procurador (terceira parte) na rede Peer-to-Peer. A estação da rede interna informa ao procurador, o qual possui um endereço IP público, o seu conjunto de recursos. Quando um elemento deseja um dos recursos disponibilizados pelo elemento localizado atrás do *firewall*, o procurador responde com uma resposta positiva. Então, o nó que lançou a requisição tenta abrir uma conexão com o procurador. O procurador, por sua vez, é continuamente acessado pela estação com endereço IP privado (lembre-se que a rede utiliza o NAT), o qual questiona a ele se nenhum dos seus recursos foi requisitado. Nesse caso, o procurador responde que sim e informa o endereço do elemento que deseja e o nome do recurso. Nesse instante, a estação interna conecta-se ao elemento indicado e, através de uma técnica *push*, "empurra" o recurso para a outra ponta da conexão.

Como todo sistema distribuído, as redes Peer-to-Peer necessitam um esquema de endereçamento. O sistema de resolução de nomes da Internet (DNS) não se encaixa nos requisitos apresentados pelas redes P2P. Isso acontece principalmente porque ele foi desenvolvido para redes que mudam pouco (uma estação de trabalho não muda de nome diariamente). Os sistemas P2P são mais dinâmicos e exigem esquemas alternativos para nomeação de seus elementos. Técnicas que utilizam certificados digitais e uma infraestrutura de chaves públicas estão entre aquelas mais utilizadas para esse fim.

A forma como a aplicação descobre os recursos na rede Peer-to-Peer é uma de suas principais características. Os modos mais comuns de busca e roteamento são o centralizado, por inundação e o modelo que usa uma tabela de hash distribuída (DHT) [ZHU 05, BUE 03]. Na construção de um sistema P2P puro pode-se optar pelas duas últimas opções.

[3]O SOAP e o XML são protocolos utilizados essencialmente na comunicação entre *Web Services*. Eles são padronizados pela W3C (*World Wide Web Consortium*).

O modelo de inundação é o mais simples de implementar, porém apresenta alguns problemas quanto a escalabilidade. A Figura 2.7 mostra esse modelo. Como exposto na seção 2.2 (Classificação das Redes Peer-to-Peer), no algoritmo de inundação cada nó envia a requisição que recebe para todos os seus vizinhos, até que algum deles possui o recurso e atende a requisição com uma resposta positiva.

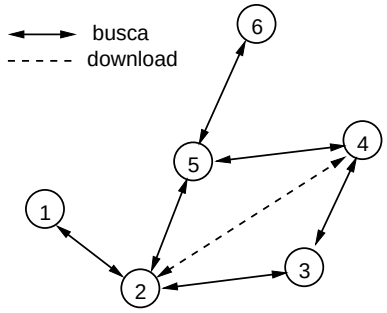

Figura 2.7: Modelo de encaminhamento de mensagens por inundação

Uma das principais fontes de ineficiência da busca em redes não estruturadas é a falta de informações sobre a distribuição dos dados na rede [DET 03]. O modelo DHT possibilita essa estruturação. Nele, todo elemento adquire um identificador único (chave) ao entrar na rede Peer-to-Peer. Quando um documento é publicado no sistema, também lhe é designado um identificador único; calculado a partir de seu nome e conteúdo. As chaves atribuídas aos elementos da rede Peer-to-Peer e aos recursos inseridos nela estão no mesmo espaço de chaves, isto é, possuem o mesmo número de dígitos.

Logo após a inclusão do recurso, ele é roteado (enviado) até o elemento da rede que possui o identificador mais similar ao seu. Para isso, cada nó da rede, ao receber o recurso publicado, verifica qual de seus vizinhos possui o identificador mais próximo daquele designado ao recurso e encaminha-o a esse elemento (cada nó observa também o seu próprio identificador nesse processo). Em todos os algoritmos que utilizam a DHT existe uma função de proximidade entre a chave publicada (ou procurada) e a chave concedida aos elementos do sistema.

Percebe-se na Figura 2.8 que o recurso com identificador 0008 foi roteado até o elemento representado como 0010. Durante o processo de roteamento, uma cópia local

do recurso é realizada em cada nó intermediário. Quando um elemento deseja encontrar um recurso na rede P2P, ele lança a requisição na rede (na forma de identificador) e ela é encaminhada até o nó com identificador mais semelhante ao encontrado na solicitação. Se esse elemento contém o recurso desejado, ele emite uma mensagem de resposta que chega até o nó origem e o informa do sucesso da busca.

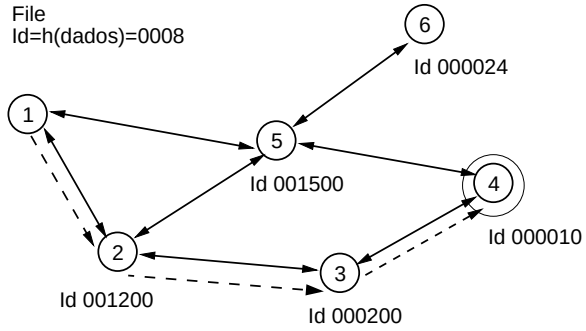

Figura 2.8: Modelo de roteamento DHT

Como se pode constatar, o usuário precisa conhecer o identificador do recurso antes de lançar uma requisição na rede. Esse é um dos motivos que tornam o modelo DHT mais complexo que o modelo de inundação. Uma alternativa para essa situação é a utilização da Web para divulgar o nome dos recursos contidos na rede e seus respectivos identificadores.

2.4 Aplicações Peer-to-Peer

A quantidade de aplicações Peer-to-Peer existentes na Internet é grande. Essas aplicações estão divididas em três grupos [BAR 01]: *i*) computação distribuída; *ii*) compartilhamento de arquivos; *iii*) trabalho colaborativo. A primeira classe de aplicações P2P envolve os sistemas criados para a resolução de problemas computacionalmente difíceis. Seu principal integrante é o sistema SETI@Home, o qual utiliza os ciclos de processamento (CPU) dos computadores localizados na borda da rede para encontrar algum sinal que evidencie a presença extraterrestre no espaço.

Esse sistema contém um elemento central (mestre) responsável por enviar para

cada integrante da rede (escravos) um conjunto de informação que este deve processar. Os escravos processam individualmente seus pedaços de informação e retornam ao mestre o resultado do processamento. O mestre armazena os resultados que lhe são enviados, analisa-os e verifica se algum elemento da rede obteve sucesso na procura por sinais extraterrestres. Essa arquitetura de distribuição possibilitou ao SETI@Home[4] um poder computacional de 52.75 TeraFlops/segundo (trilhões de operações de ponto flutuante por segundo) [SAD 04].

A classe de aplicações P2P responsável pelo compartilhamento de arquivos[5] trouxe popularidade aos sistemas Peer-to-Peer. Alguns exemplos de aplicações P2P utilizadas para a troca de recursos são o Napster, Gnutella, Freenet e KazaA.

O terceiro grupo engloba aqueles sistemas onde os usuários que estão na borda da rede comunicam-se em tempo real, geralmente com algum objetivo em comum. As aplicações para o trabalho colaborativo diferenciam-se das classes anteriores, pois apresentam a interação direta entre pessoas (*People-to-People*), ao invés de apenas comunicações entre aplicações. O potencial desse gênero de computação Peer-to-Peer é imenso, desde a edição simultânea de documentos até a criação de grupos de discussão e colaboração em tempo real. Nesses sistemas, todos os elementos devem possuir a mesma visão dos acontecimentos ocorridos no decorrer da comunicação. Essa exigência requer a utilização de métodos de sincronização na rede formada. Assim, sempre que mudanças ocorrem no ambiente de colaboração, são lançados eventos que informam aos interessados o novo estado do grupo.

Este documento seleciona duas aplicações P2P e descreve-as em detalhes. O objetivo é transmitir o funcionamento de um sistema Peer-to-Peer típico, desde a divulgação de recursos na rede até os procedimentos necessários para o encontro de informações no sistema. São apresentados, na seqüência, os sistemas Gnutella [ADA 00] e Freenet [CLA 01].

[4] `http://setiathome.ssl.berkeley.edu/`

[5] Os arquivos trocados contém textos, músicas, gráficos, vídeos, programas, fotos, etc.

2.4.0.1 Gnutella

O Gnutella foi o primeiro protocolo Peer-to-Peer totalmente descentralizado na Internet (surgiu em meados de 2000) [MIL 04]. Ele também é referenciado como uma linguagem de comunicação entre *Peers* e, por adotar paradigmas inovadores, sua relevância é alta no grupo de aplicações Peer-to-Peer atuais. No Gnutella um nó que deseja conectar-se a rede precisa encontrar um ponto de entrada (ou de acesso) para o sistema. Cada membro da rede pode servir como ponte para que outros nós se unam ao ambiente colaborativo.

A Tabela 2.2 exibe as etapas existentes no processo de adição de um novo participante ao Gnutella. Percebe-se que esse novo participante cumprimenta um dos membros da rede e este apresenta o novo integrante aos seus conhecidos. A partir desse instante a rede Gnutella reconhece o novo elemento e é reconhecida por ele.

Número	**Descrição**
1	Descoberta de um ponto de acesso (porta de entrada) à rede Gnutella
2	Emissão de uma mensagem PING ao ponto de acesso escolhido
3	O elemento ponto de acesso reconhece o pedido do novo participante e repassa o seu PING para a sua vizinhança (podem ser centenas de nós)
4	Os nós que recebem o PING enviam uma mensagem PONG informando que passam a conhecer o novo integrante do ambiente
5	O elemento que deseja entrar no Gnutella recolhe as mensagens PONG e monta sua lista de estações conhecidas
6	No final, o novo elemento é agregado ao Gnutella e torna-se parte ativa da topologia

Tabela 2.2: Etapas da conexão de um elemento ao Gnutella [ORA 01]

No momento que um nó deseja uma informação, ele elabora uma mensagem de requisição e envia para os elementos que ele conhece (aqueles que lhe foram apresentados). Cada nó que recebe um pedido vindo da rede verifica se possui o item requerido e, caso possuir, responde positivamente à solicitação. O Gnutella especifica que a mensagem

de requisição, independente da resposta fornecida pelo nó que a recebeu, deve prosseguir na rede Peer-to-Peer. Essa característica possibilita a busca em profundidade no sistema.

O Gnutella precisa tratar duas circunstâncias que podem ocorrer com a mensagem de solicitação de recurso. A primeira delas é referente ao problema ocasionado pela super-conectividade dos nós da rede Peer-to-Peer, onde uma mensagem desse tipo pode chegar mais de uma vez ao mesmo elemento. O Gnutella determina que cada mensagem de requisição possui um identificador único (UUID) de 128 *bits*. Os nós devem armazenar por algum intervalo de tempo a identificação das mensagens que repassam para a rede. Essa técnica evita o envio duplicado de uma mesma mensagem.

Outra questão que o Gnutella enfrenta é como as mensagens de requisição deixam o sistema. Elas não devem permanecer para sempre no ambiente Peer-to-Peer. Ele supre essa necessidade com a utilização de um campo "tempo de vida" (TTL[6]) no cabeçalho da mensagem. Assim, o pedido se propaga até determinado nível (profundidade a partir do nó origem) da rede e então é descartado. Essa peculiaridade é uma das justificativas para o fato de dois elementos receberam respostas diferentes do Gnutella, lançando a mesma chave de busca. No Gnutella o valor do TTL é 7.

A resposta positiva de um membro da rede Gnutella não é transmitida diretamente para o elemento que originou o chamado. Ela faz o caminho inverso da mensagem de requisição, passando por alguns nós intermediários até chegar ao destino (local da procedência da procura). Esse fato ocorre porque o nó que responde a uma solicitação não sabe qual o endereço IP do nó que gerou a busca na rede. Ele conhece apenas o identificador único da mensagem (UUID) e o endereço do nó que o repassou a requisição.

Como mencionado, todos os nós da rede Gnutella armazenam, por um breve período de tempo, a identificação das mensagens de requisição que chegam até eles. Então, quando determinado elemento recebe uma mensagem-resposta, ele extrai o seu identificador e encontra, entre as mensagens que guarda (*cache*), aquela que possui o mesmo identificador. Desta maneira, ele saberá para qual endereço deve enviar a mensagem-resposta a fim dela cada vez mais se aproximar de seu destino (emissor da procura).

Esse processo é conhecido como roteamento inverso. Em certo momento a men-

[6]O TTL tem uma semântica análoga à utilizada no cabeçalho do pacote IP, mas em nível de aplicação.

sagem-resposta irá chegar até ao seu destino – o elemento que lançou a procura na rede Gnutella. Esse nó conhece qual é o identificador UUID que acompanhou a sua requisição, uma vez que foi ele quem o gerou. Sendo assim, quando ele recebe uma mensagem-resposta contendo esse mesmo identificador, percebe que o roteamento inverso chegou ao fim. Então, extrai o endereço IP do elemento P2P detentor do recurso almejado e abre uma conexão direta com ele.

No princípio do Gnutella as estações precisavam encontrar o endereço de um ponto da rede para poderem unir-se ao sistema Peer-to-Peer. As páginas Web e os servidores de IRC mantinham endereços de nós permanentemente conectados ao Gnutella e que poderiam ser utilizados como pontos de acesso. Nesse esquema, a parte da rede que um determinado nó se conectava era aleatória e o sistema possuía um bom índice de balanceamento.

A descoberta manual de um ponto de conexão no Gnutella tornou-se impopular, o que favoreceu o aparecimento dos elementos chamados “cache de hosts”. Esses elementos oferecem uma maneira fácil de o usuário participar do Gnutella. Eles cumprimentam o novo nó e passam a ele o conjunto de identificadores de outros nós da rede. A Figura 2.9 expõe uma topologia onde os “cache de hosts” estão presentes.

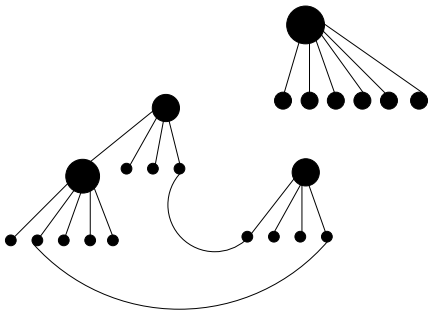

Figura 2.9: Topologia da rede Gnutella com *Cache de Hosts*

O uso extensivo dos “caches de hosts” pode ocasionar alguns problemas à rede Gnutella. Há uma grande concentração de nós em torno desses elementos e congestionamento em porções específicas da rede. O motivo principal para essa situação é o envio, por determinado “cache de hosts”, de uma lista igual de nós conhecidos a todo interessado

que o procura.

O Gnutella utiliza um mecanismo pseudo-anônimo em suas operações. As mensagens de requisição não revelam qual estação lançou o chamado na rede. Além disso, o sistema de roteamento Gnutella não é acessível externamente. Porém, quando um membro realiza um *download* de um recurso de outro integrante, ele informa o seu endereço IP a outra ponta da conexão. O endereço IP do fornecedor da informação também fica descoberto.

O Gnutella é vulnerável aos nós caronas – aqueles que não compartilham recursos e atuam como sanguessugas na rede. Ele não possui mecanismos para diferenciar quais nós são responsáveis no ambiente colaborativo. Os elementos caronas receberão pedidos e sempre irão repassá-los à frente; nunca respondê-los positivamente. Essa atitude apenas aumenta o tempo de envio de uma solicitação e contribui para a degradação do desempenho do sistema em geral.

2.4.0.2 Freenet

O Freenet é um sistema descentralizado de distribuição de arquivos que possui como principal característica garantir a anonimidade nas comunicações entre os elementos da rede [CLA 01, CLA 02]. Os seus principais objetivos são permitir que pessoas distribuam recursos anonimamente, permitir consultas anônimas ao conjunto de recursos e impossibilitar a retirada completa de um recurso da rede por algum órgão de censura ou entidade imprópria. O uso extensivo de técnicas de replicação é uma das formas encontradas pelo Freenet para efetivar os seus objetivos.

O Freenet constrói um grande repositório de informações, onde cada nó da rede Peer-to-Peer colabora com recursos e espaço de armazenamento. Diferente do Gnutella que prevê que cada nó conhece claramente os recursos que está compartilhando para o ambiente, no Freenet os recursos circulam entre os discos rígidos ociosos e cada elemento não possui a identificação do conteúdo que está guardado em seu disco em dado instante (as meta-informações sobre os recursos e eles próprios estão cifrados em cada local de armazenamento). Ele entende apenas que participa e colabora com o Freenet.

O Freenet é uma rede *overlay*, pois desenvolve uma topologia virtual em cima da Internet. Os nós unem-se ao Freenet em um esquema semelhante àquele observado no Gnutella, ou seja, através de um nó conhecido que lhe serve como porta de entrada para o sistema. Todos os elementos da rede estabelecem, no momento que se unem à topologia, a quantidade de disco rígido que colaborarão para a criação do sistema de armazenamento estipulado pelo Freenet. A solicitação de conteúdo e o envio das respostas aos requisitores são semelhantes no Freenet e no Gnutella. No entanto, existem algumas diferenças, principalmente no modo de difusão da mensagem de procura por recursos.

A resposta bem sucedida é roteada pelo mesmo caminho da solicitação. A diferença é que os membros intermediários também armazenam o recurso pesquisado por algum intervalo de tempo. Dessa forma, caso houverem novas solicitações pelo mesmo recurso, as respostas são entregues mais rapidamente, pois estarão mais próximas dos nós que as requerem. A replicação dos recursos na topologia possibilita também que o Freenet seja mais tolerante a falhas que outros projetos Peer-to-Peer.

Cada membro da rede Freenet armazena uma quantidade limite de *bytes*. É utilizado o sistema de pilhas para conhecer quais recursos têm mais importância (mais solicitados) em cada elemento. Nele os recursos mais acessados estão no topo da pilha e, conseqüentemente, devem permanecer mais tempo no repositório. Os recursos que passam muito tempo sem ser consultados são excluídos da rede. A Figura 2.10 apresenta uma pilha exemplo em um elemento Freenet.

	Chave	Informação	Endereço
1	jkasjkasiusdkkas	sdlkso9s9s	tcp/endereçoX
2	djasjkasjkasjksjk	xjksjksksk	tcp/endereçoY
3	asklaslao96790	abbaallallu	tcp/endereçoZ
4	a78alanaia9m1y	abnalau818	tcp/endereçoW

Figura 2.10: Organização do repositório em um elemento Freenet

O Freenet utiliza o modelo de roteamento representado na Figura 2.8 (seção 2.3). Nesse sistema as duas operações possíveis são a inserção de um arquivo e a procura por

um item da rede. No processo de inserção, como explicado na seção 2.3, é designado um identificador ao novo recurso e ele é roteado até o elemento que possui a identificação mais próxima a sua. Esse fato possibilita a estruturação da rede e facilita o processo de busca[7]. Além disso, todos os elementos mantêm individualmente uma tabela com ponteiros para outros recursos. Os membros da rede constroem essa tabela pois conseguem aprender com as requisições que repassam e com as mensagens-resposta que enviam pelo roteamento inverso.

Outra diferença entre os sistemas Peer-to-Peer descritos nessas duas seções é a maneira como o Freenet realiza o roteamento das mensagens de solicitação. No Gnutella, quando um nó recebe um pedido, ele o avalia e logo após repassa para todos os seus conhecidos (podem ser centenas de outros participantes). Por sua vez, o Freenet usa uma difusão direcionada; ele encaminha a solicitação para apenas um nó por vez na rede (*unicast*). Quando um nó recebe uma requisição, ele extrai o identificador do recurso procurado e verifica no seu repositório se o possui (resposta positiva). Caso não possuir, ele verifica qual de seus vizinhos possui o identificador mais próximo ao da solicitação e encaminha o pedido a esse elemento.

Diferentemente do Gnutella, no Freenet não existe a recuperação direta de um recurso, ou seja, inexiste a comunicação Peer-to-Peer entre o lançador da requisição e o detentor do recurso. O recurso, da mesma forma que a mensagem-resposta, deve percorrer nós intermediários até chegar ao destino. Essa particularidade reforça o enfoque dado pelo Freenet ao tema anonimidade.

O poder de escolha da rota mais próxima para atender a uma solicitação é a principal característica do Freenet. Como cada nó procura encaminhar a solicitação para cada vez mais perto dos dados, a busca é muitas vezes mais poderosa que uma busca linear e muito mais eficiente que uma difusão descontrolada [CLA 01].

Os nós caronas não são um problema para o ambiente Freenet. Ele resolve esse problema da seguinte maneira: os nós caronas não compartilham recursos e não respondem positivamente às solicitações. Conseqüentemente, nenhum outro nó ganhará re-

[7]No Freenet recursos com identificadores (chaves) próximos tendem a ser roteados até o mesmo elemento.

ferência a ele e, para a rede Freenet, é como se esse nó não existisse (lembre-se do direcionamento das solicitações). Porém, suas consultas seguirão sendo roteadas no Freenet e, assim, ele estará auxiliando para o consumo de largura de banda do sistema Peer-to-Peer.

2.5 Gerência de Tráfego Peer-to-Peer

A utilização de redes Peer-to-Peer, principalmente para o compartilhamento de arquivos, cresceu drasticamente nos últimos anos [IZA 04]. O uso abusivo destas aplicações acarreta no alto consumo da largura de banda das organizações, ocasionando um impacto negativo na rede. Com o intuito de amenizar os problemas surgidos com o compartilhamento de arquivos através de comunicações Peer-to-Peer, diversas metodologias foram propostas na literatura, tanto para identificar o tráfego, como para controlá-lo [SEN 04, IZA 04].

O primeiro desafio para se poder aplicar um controle de tráfego Peer-to-Peer efetivo é a sua identificação. O trabalho de Leibowitz [LEI 02] desenvolve uma arquitetura de *cache* P2P onde um elemento, de forma transparente, intercepta todas as conexões P2P da organização e verifica, antes de abrir uma conexão externa, se não atende diretamente a requisição deste cliente P2P. Esta abordagem consegue bons resultados, porém ela possui o seguinte ponto negativo: o componente que realiza o *cache* identifica as comunicações P2P baseado apenas nos números de porta padrões das aplicações de compartilhamento de arquivos mais comuns.

O trabalho de Karagiannis [KAR 03] afirma que entre 30 e 70% do tráfego Peer-to-Peer não é reconhecido apenas pelo monitoramento das portas padrões das aplicações P2P. A pesquisa [KAR 03] estudou o comportamento das diversas redes P2P e formulou heurísticas capazes de detectar as comunicações Peer-to-Peer simples e sofisticadas – aquelas que utilizam mecanismos diferenciados como o transporte de dados sobre o protocolo HTTP para sobrepor os limites impostos por *firewalls* e outros mecanismos de proteção.

A pesquisa de Hamada et al. [HAM 04] concentrou-se em analisar e identificar o tráfego da rede Gnutella. Ele utilizou simuladores de rede para recriar os padrões e as

características do tráfego Peer-to-Peer e conclui que, na maioria das vezes, 40% da largura de banda de um backbone de alta velocidade é desprendida com as comunicações Peer-to-Peer. Broido et al. [BRO 04] reafirma a necessidade de utilizar meios mais robustos para identificar o tráfego P2P, já que a análise baseada apenas em portas não permite a identificação completa desse tipo de tráfego. Segundo [BRO 04], para reconhecer o tráfego P2P com clareza deve-se investigar o campo de dados de cada pacote que é transmitido na topologia de rede da organização.

Outra pesquisa que procura gerenciar o tráfego Peer-to-Peer é [RIG 04b]. Ela desenvolve uma arquitetura chamada P2P-Limit[8] capaz de identificar, controlar e monitorar este paradigma de comunicação. Seus principais objetivos são reduzir as comunicações P2P de uma organização, otimizar o uso da largura de banda e contabilizar de forma concreta este tipo de tráfego na topologia de rede. A principal decisão de projeto do P2P-Limit é a opção pelo não bloqueio das aplicações P2P (apenas redução e controle), já que elas possuem mecanismos apurados para sobreporem as barreiras impostas pelos administradores de redes. Destacam-se nesta arquitetura os elementos responsáveis pela contabilização dos pacotes Internet transmitidos na topologia, representado pelo NetFlow, e a identificação do tráfego P2P em trânsito no perímetro de rede analisado, designado pelo detector de intrusos (IDS) Snort.

2.6 Balanço

O capítulo 2 preocupou-se principalmente em mostrar as características e o modo de funcionamento dos sistemas Peer-to-Peer. A comunicação Peer-to-Peer é bastante diferente do padrão cliente-servidor conhecido. Nas redes P2P os nós agem como clientes e servidores simultaneamente, o tráfego é simétrico (não apenas em um sentido) e todos os integrantes do ambiente executam o mesmo programa (código-fonte).

De uma forma geral, as redes P2P podem ser divididas em puras e híbridas. As redes P2P puras são totalmente descentralizadas e não necessitam de elementos centralizados. As redes P2P híbridas, por sua vez, precisam de um ou mais pontos centralizados

[8] `http://www.pop-sc.rnp.br/site/p2p`

para dar suporte à computação colaborativa P2P. Estes elementos (centrais), na maioria das vezes, possuem a função de catalogar os recursos da rede P2P e em quais nós eles podem ser encontrados.

As aplicações P2P emergiram rapidamente e a gerência e o controle desse tipo de tráfego tornou-se indispensável nas organizações. As redes P2P trouxeram um novo modelo de distribuição de informações e muitos benefícios para os usuários leigos da Internet. No entanto, o crescimento do volume de comunicações Peer-to-Peer auxilia para o aumento do congestionamento nos enlaces de rede e para a perda de desempenho nas demais conexões dos usuários. Deve-se, sempre que possível, encontrar um ponto médio entre as vantagens e desvantagens dos sistemas P2P, de forma que ambos usuários e administradores de redes estejam satisfeitos.

Os modelos cliente-servidor e Peer-to-Peer não são mutuamente exclusivos, isto é, eles podem existir (e funcionar) juntos em um mesmo segmento ou topologia de rede. Algumas aplicações irão se adaptar melhor ao modelo de comunicação proposto pelo paradigma Peer-to-Peer, enquanto outras irão utilizar o padrão cliente-servidor difundido em larga escala na Internet.

Capítulo 3

JXTA e P2PSockets

O objetivo deste capítulo é descrever de forma sintética os projetos JXTA e P2P-Sockets. Eles estão relacionados ao desenvolvimento de sistemas Peer-to-Peer e foram utilizados no processo de pesquisa descrito nesta dissertação de mestrado. São exibidos os objetivos, benefícios e funcionamento de cada um destes projetos.

3.1 Tecnologia JXTA

O objetivo principal do projeto JXTA[1] [Sun 04] é prover uma plataforma que contenha as funções básicas necessárias para o desenvolvimento de aplicações Peer-to-Peer. Ele especifica um conjunto de protocolos que permitem que membros de uma rede virtual comuniquem-se e colaborem uns com os outros.

O desenvolvimento de aplicações distribuídas requer a elaboração de uma infraestrutura de comunicação eficiente e robusta. A realização desta estrutura consome tempo, recursos e, em determinados casos, exige uma equipe de profissionais. O JXTA fornece uma infraestrutura de comunicação P2P completa e possibilita, assim, que organizações e instituições de ensino concentrem seus focos na criação de aplicações Peer-to-Peer e não inventem ou reescrevam uma camada de comunicação novamente.

O JXTA procura corrigir alguns aspectos negativos encontrados em outros proje-

[1] `http://www.jxta.org`

tos Peer-to-Peer como a falta de interoperabilidade entre aplicações e a independência de plataforma. A tecnologia JXTA pode executar em vários tipos de dispositivos, incluindo telefones celulares, computadores portáteis, sensores eletrônicos e supercomputadores [YEA 02]. Estes dispositivos podem interagir na rede virtual JXTA independente de suas posições, ambiente de operação ou mesmo quando estiverem em redes protegidas por *firewalls*, tradutores de endereço de rede (NAT) ou que aplicam protocolos de transporte diferentes.

O JXTA padroniza um conjunto de protocolos e componentes (blocos fundamentais), os quais formam o núcleo da terminologia deste projeto. Os principais componentes são os nós, grupos, serviços, canais, anúncios e mensagens[2] [HAL 03]. A rede JXTA consiste de uma série de nós interconectados que podem se auto organizar em grupos. Os grupos disponibilizam uma série de serviços a seus integrantes, como o compartilhamento de recursos e a troca de mensagens em tempo real. Os nós da rede anunciam os seus serviços através de documentos XML chamados anúncios. Estes documentos possibilitam que outros membros da rede aprendam como conectar e interagir com os serviços disponibilizados pelo nós e pelos grupos do ambiente colaborativo. Os canais, por sua vez, são usados para o transporte de mensagens entre os nós. Elas são representadas através do protocolo XML e podem conter informações de roteamento, textos simples, arquivos binários e outros.

Segundo Wilson [WIL 02], existem três tipos de nós em uma rede JXTA. O primeiro é denominado nó simples e não possui muitas responsabilidades administrativas; ele apenas serve e consome recursos. O segundo é chamado de nó *rendezvous* (local de encontro) e suas funções fundamentais são manter um histórico dos anúncios de recursos publicados na rede e repassar as requisições de procura para outros nós *rendezvous*, a fim de auxiliar os nós da topologia na descoberta de recursos. Os nós *rendezvous* mantém uma lista de nós conhecidos (e outros recursos JXTA) e os demais elementos da rede requisitam a ele esta informação. Os nós roteadores[3] compõem a terceira classificação. Eles são utilizados na comunicação entre dois nós que não podem se comunicar diretamente

[2] *Peers groups, services, pipes, advertisements, messages.*

[3] Os nós roteadores também são designados *Relay Peers*.

(por exemplo, quando um deles possui endereço de rede privado). Portanto, eles realizam a função de ponte no processo de comunicação Peer-to-Peer. Mais de um nó roteador pode estar presente entre dois extremos que não conseguem se comunicar de forma direta. Cada nó da rede JXTA, independente de seu tipo (um nó pode incorporar vários tipos), possui um identificador universal único.

O conceito de grupo é relevante no JXTA pois permite dividir a rede em conjuntos menores. Um nó pode acessar os recursos e serviços localizados apenas nos grupos que pertence e as requisições de procura emitidas por ele não são propagadas para grupos dos quais não é membro. O JXTA define um núcleo de serviços que todo o grupo deve oferecer. Dois exemplos são os serviços de descoberta e de controle de participantes. O serviço de descoberta é utilizado pelos nós para encontrar os recursos localizados no grupo. São considerados recursos na rede JXTA os nós, grupos, canais, serviços, entre outros; eles são publicados através da divulgação de anúncios e encontrados pelos interessados. Já o serviço de controle de participantes é usado para determinar quais usuários são aceitos ou negados no grupo e as exigências que precisam ser cumpridas por um elemento para ter seu acesso ao grupo autorizado.

Todos os recursos na rede JXTA são representados por anúncios. Os anúncios são metadados estruturados como documentos XML. Os protocolos JXTA utilizam os anúncios para descrever e publicar a existência de recursos. Os nós descobrem os recursos através da pesquisa e descoberta de seus respectivos anúncios. O anúncio de um nó contém informações específicas deste participante da rede, como seu nome, seu identificador, seus pontos de acesso e outros atributos relevantes para o grupo Peer-to-Peer. Outro tipo de anúncio é aquele que divulga um nó *rendezvous*; nele é informado à comunidade Peer-to-Peer que determinado membro está atuando como nó *rendezvous*.

O JXTA define uma série de formatos de mensagens XML, ou protocolos, para a comunicação entre os nós da rede virtual. Cada protocolo do JXTA é responsável por uma função específica na rede Peer-to-Peer. Os protocolos padronizam, por exemplo, a forma como os nós são descobertos, como se obtém informações de estado de um nó, qual o conjunto de etapas necessárias para invocar um serviço na rede, qual o procedimento para criar, se ligar e sair de grupos P2P, como abrir e sustentar conexões e como rotear

mensagens através de outros nós.

Existem seis protocolos no JXTA. A Figura 3.1 apresenta seus nomes, siglas e principais funções. O PDP (*Peer Discovery Protocolo*) está associado ao serviço de descoberta e possibilita que os nós procurem por anúncios na rede. A verificação do estado de um nó Peer-to-Peer (por exemplo, verificar se ele está ativo ou não) é uma das funções principais do PIP (*Peer Information Protocol*). O PRP (*Peer Resolver Protocol*) é utilizado pelos nós P2P para enviar requisições variadas a outros membros da rede e identificar as respostas recebidas. O PRP usa o serviço de *Rendezvous* para disseminar uma requisição para múltiplos nós. Os protocolos PIP e PDP são construídos utilizando-se o PRP.

O PBP (*Peer Binding Protocol*) especifica um mecanismo para associar (amarrar) um canal de comunicação virtual (*pipe*) com um ponto de acesso (por exemplo, portas TCP). Este protocolo é utilizado para estabelecer e manter canais virtuais entre os nós. Por sua vez, o ERP (*Endpoint Routing Protocol*) define um conjunto de requisições e consultas que são usadas para encontrar informações de roteamento. Estas informações são necessárias para a transmissão de mensagens entre os integrantes do sistema.

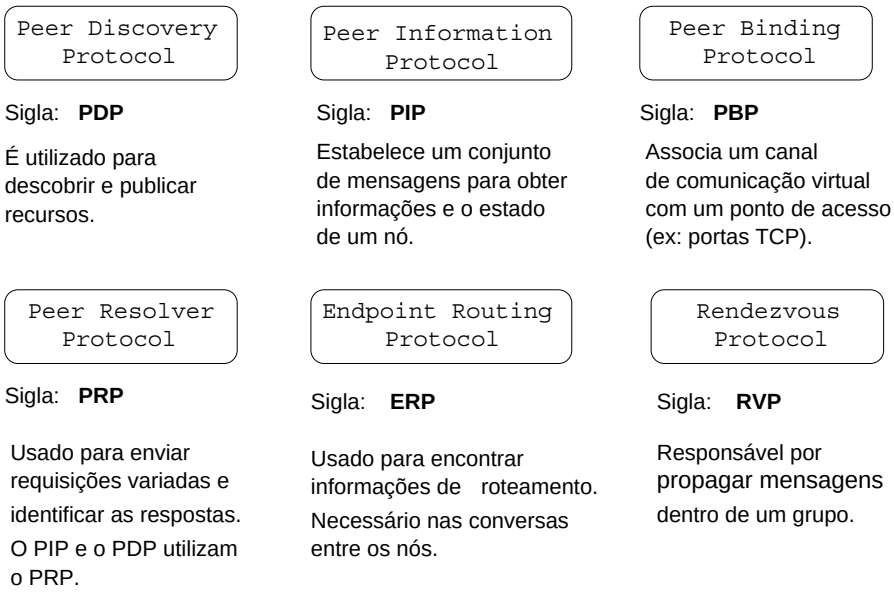

Figura 3.1: Protocolos do projeto JXTA

O último protocolo chama-se RVP (*Rendezvous Protocol*), o qual é responsável por propagar mensagens dentro de um grupo. O RVP também coordena como são feitos os repasses de mensagens entre os nós *rendezvous* e a detecção *loops* de encaminhamentos entre esses nós. Todos os protocolos discutidos são assíncronos e baseiam-se no modelo consulta/resposta. Os nós da rede não precisam executar todos os seis protocolos; eles somente necessitam adotar aqueles que irão utilizar.

A Figura 3.2 exibe a arquitetura do projeto JXTA. A camada mais inferior denomina-se Núcleo JXTA e encapsula os componentes essenciais necessários a toda aplicação Peer-to-Peer. Fazem parte desta camada os protocolos exibidos na Figura 3.1, o mecanismo de identificação de nós e os blocos fundamentais (especificação dos componentes básicos) do JXTA. A camada de Serviços inclui os serviços da rede Peer-to-Peer que são desejados, porém não imprescindíveis. Os serviços implementam funcionalidades como o compartilhamento de recursos, sistemas de arquivos distribuídos e infraestrutura de chaves públicas (PKI). A camada de Aplicações é a mais superior. Nela estão implementadas as aplicações Peer-to-Peer completas, como os sistemas de leilão distribuídos.

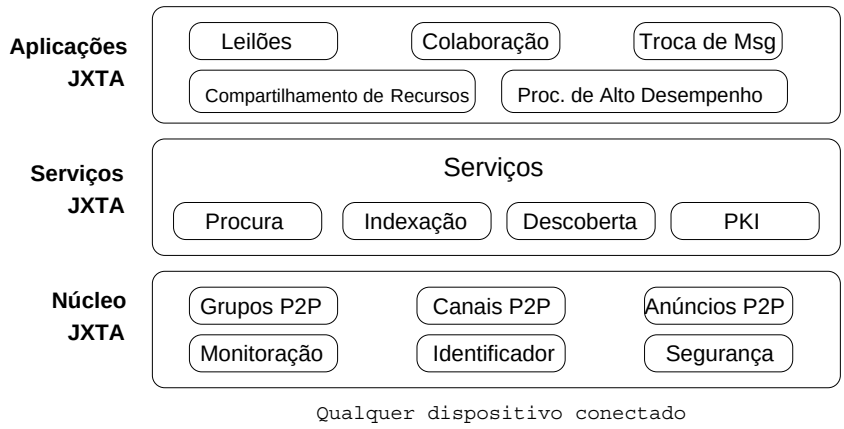

Figura 3.2: Arquitetura do projeto JXTA

Existem três maneiras básicas de um nó Peer-to-Peer encontrar anúncios e, conseqüentemente, descobrir os recursos de um rede JXTA. O caminho mais fácil para um nó descobrir anúncios é pesquisar a sua base de anúncios antigos. Este método não exige envio de mensagens, porém o nó possui o risco de conhecer anúncios ultrapassados –

aqueles que descrevem recursos que não estão mais disponíveis no ambiente colaborativo. A segunda técnica chama-se Descoberta Direta e nela o nó que deseja conhecer anúncios espalha uma requisição de procura em sua rede local (*broadcast* ou *multicast*). Todos os demais nós da rede local respondem ao requisitor. Quando um nó responde a uma mensagem de descoberta, ele envia informações sobre si mesmo e sobre os demais nós que ele descobriu anteriormente. Se houver um nó *rendezvous* na rede local, ele observará, como os demais dispositivos da LAN, a difusão direcionada e repassará a mensagem de descoberta aos nós que ele conhece (a mensagem poderá ir além da rede local). Caso não exista na rede local um nó *rendezvous*, somente nós e recursos do mesmo segmento de rede do requisitor serão descobertos. Esse procedimento é limitado aos nós localizados no mesmo segmento de rede e normalmente não pode ser utilizado para descobrir nós externos a sua rede.

A terceira técnica denomina-se Descoberta Indireta e requer a utilização de um nó *rendezvous*, o qual atua como fonte de anúncios. Essa técnica pode ser utilizada por nós de uma rede local para encontrar anúncios sem a necessidade de difusão de mensagens e por aqueles nós situados em uma rede privada para encontrar nós e recursos fora de seu perímetro de rede.

O nó *rendezvous*, logo após receber a requisição por anúncios, repassa a mensagem para os nós que ele conhece, incluindo outros nós *rendezvous*. Da mesma forma que um nó simples, o nó *rendezvous* armazena localmente os anúncios que percebe na rede Peer-to-Peer. Ele utiliza esta base de anúncios para responder ao requisitor. Para aqueles nós localizados em rede privadas, protegidas ou que adotam tradutores de endereços de rede, encontrar um nó rendezvous e um nó roteador é crítico para que eles participem do ambiente JXTA. Devido às restrições de comunicação impostas por essa classe de redes de computadores, um nó da rede interna não tem a capacidade de usar a Descoberta Direta para encontrar nós externos a sua rede. No entanto, esse nó pode executar a Descoberta Indireta através de um nó *rendezvous* e um nó roteador. Na maioria das aplicações P2P desenvolvidas com o JXTA, a melhor forma de garantir que um nó simples encontre nós *rendezvous* e roteadores é fornecer junto com a aplicação P2P um conjunto de endereços IP (estáticos e públicos) desses nós especiais.

A Descoberta Direta é totalmente dinâmica, ou seja, não exige nenhuma configuração especial no participante Peer-to-Peer. Já na técnica de Descoberta Indireta está presente necessariamente a figura do nó *rendezvous*. Os nós *rendezvous*, como mencionado, podem ser descobertos através de uma configuração (manual) na própria aplicação P2P ou através da percepção de um anúncio na rede JXTA que publica a existência de nós que atuam como *rendezvous*. Sendo assim, um elemento da rede que utiliza o protocolo PDP e as técnicas anteriores para a descoberta de anúncios, percebe os recursos dispostos em sua rede local e aqueles conhecidos pelos nós *rendezvous* com quem entrou em contato.

A comunicação de um nó simples com um nó *rendezvous* pode acontecer através dos protocolos TCP/IP ou HTTP. O transporte de mensagens sob o HTTP possibilita que nós situados em redes protegidas possam sobrepor barreiras como *firewalls* e estabelecer comunicações bidirecionais (requisição/resposta) com nós *rendezvous*. A comunicação utilizando o TCP/IP ocorre quando não existe nenhum dispositivo que impeça a comunicação direta entre um nó simples e um nó *rendezvous*. Na Descoberta Direta apenas é requerido o uso do TCP/IP, pois é normal a comunicação direta entre elementos de uma rede local. Todo nó Peer-to-Peer da rede JXTA espera conexões em um ponto de acesso específico. Os membros da rede conhecem o ponto de acesso de outros participantes da rede no processo de descoberta (protocolo PDP), pois as mensagens de resposta recebidas nesta etapa contém o ponto de acesso utilizado por cada um deles.

Como mencionado, entre as principais características do JXTA está a possibilidade de nós localizados em redes protegidas por *firewalls* ou tradutores de endereços de rede (NAT) poderem usufruir dos benefícios da rede P2P normalmente. Esse tipo de nó utiliza protocolos permitidos pelo *firewall* (geralmente HTTP) para fazer o tunelamento de informações relevantes para fora da sua rede local. Quando uma conexão é iniciada na rede interna, é estabelecido o mapeamento (tradução) no dispositivo que realiza o NAT e o nó externo pode utilizar esse canal aberto com ele para transmitir dados para dentro da rede protegida.

O protocolo HTTP é do tipo requisição-resposta, ou seja, cada conexão HTTP envia um pedido e depois espera um resultado. O nó externo (rede pública) não consegue

se comunicar com o nó protegido espontaneamente; ele sempre necessita que o nó interno abra uma conexão com ele para se estabelecer uma conversação.

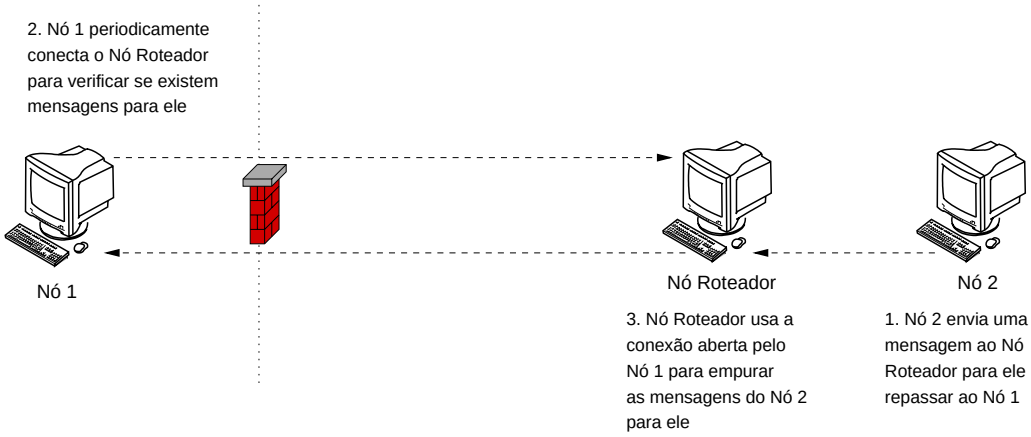

Figura 3.3: Modo de atravessar um *Firewall*/NAT

A Figura 3.3 apresenta as etapas existentes no envio de uma mensagem de um nó externo para o nó protegido. O nó externo entra em contato com o nó roteador e o nó protegido periodicamente conecta-se a esse elemento especial. Quando o nó interno abre uma conexão com o nó roteador, todas as mensagens destinadas a ele são empurradas na sua direção através de uma resposta HTTP.

Para habilitar um nó a enviar uma mensagem a outro localizado em um ambiente protegido, a origem necessita conhecer informações de roteamento que descrevem o nó roteador capaz de rotear as mensagens até o seu destino. As informações de roteamento podem ser obtidas durante o processo de descoberta ou através do uso do protocolo PRP – *Peer Routing Protocol*.

Outro cenário possível é aquele onde existem duplos *firewalls* ou NATs. A Figura 3.4 exemplifica esta situação. Antes do nó origem (1) enviar uma mensagem, ele adquire informações de roteamento que indicam o conjunto de nós roteadores que podem agir como *proxies* nesta comunicação. Quando o nó origem obtém sua informação de roteamento, o envio de mensagens envolve 4 passos:

1. a origem abre uma conexão com o seu nó roteador, requisitando a ele repassar a mensagem para o destino através do elemento roteador fornecido;

2. os nós roteadores se comunicam. Essa conexão utiliza o protocolo de transporte que eles têm em comum;

3. O roteador final espera até o destino conectar-se a ele;

4. O nó destino conecta-se ao seu nó roteador periodicamente e a mensagem é “empurrada” até ele.

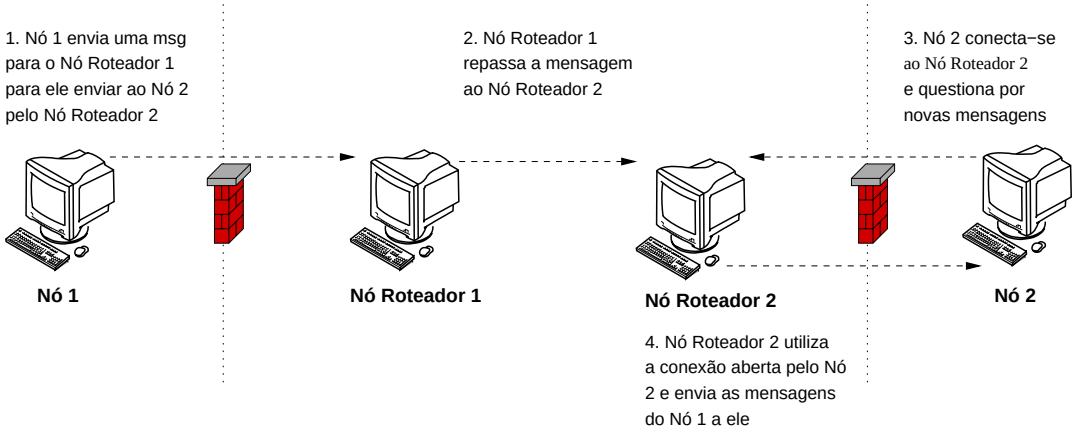

Figura 3.4: Travessia de um *Firewall*/NAT duplo

O processo de comunicação mostrado na Figura 3.4 poderia acontecer com apenas um nó roteador se ambos origem e destino tivessem em comum o mesmo nó roteador. Também é relevante ressaltar que entre as duas pontas da comunicação poderia haver mais de dois nós roteadores.

3.2 Projeto P2PSockets

O P2PSockets[4] é um projeto baseado no JXTA que reimplementa o conjunto de classes padrão para comunicação remota[5] da linguagem Java. Seu objetivo é proporcionar o desenvolvimento de aplicações Peer-to-Peer, mantendo-se a mesma interface de comunicação utilizada em sistemas construídos no paradigma cliente-servidor (TCP/IP).

[4] `http://p2psockets.jxta.org`

[5] As principais classes reimplementadas são: `Socket`, `ServerSocket`, `InetAddress`.

O projeto P2PSockets permite que desenvolvedores usufruam das vantagens do JXTA, como a garantia de comunicação independente da utilização de NAT e *firewalls*, sem estarem expostos às suas complexidades [NEU 04].

Os soquetes TCP/IP são Peer-to-Peer. Contudo, na prática, percebe-se que esta afirmação não se confirma. Isso acontece principalmente devido aos *firewalls*, dispositivos de tradução de endereço de rede e questões técnicas associadas ao serviço de nomes da Internet (DNS) que impossibilitam (ou dificultam) a existência de soquetes servidores em computadores com endereços de rede privados ou dinâmicos. Os soquetes TCP/IP dependem do serviço de nomes para traduzir nomes reconhecidos pelos usuários para endereços Internet. Esta situação também auxilia para inexistência de comunicação Peer-to-Peer real entre computadores, visto que muitos dos dispositivos de uma rede não possuem endereço de rede público e um nome fixo registrado no servidor de nomes de sua organização.

A comunicação entre nós que utilizam o P2PSockets independe do serviço de nomes tradicional da Internet e das localizações dos elementos. Além da justificativa de propiciar uma comunicação Peer-to-Peer concreta entre dispositivos, outra motivação que impulsionou o desenvolvimento do projeto P2PSockets foi a complexidade inerente ao JXTA, o que torna o entendimento e o porte de aplicações para essa tecnologia uma tarefa difícil.

Apesar de sua complexidade, o JXTA é extremamente poderoso e reconhecido. O P2PSockets fornece um meio para desenvolvedores incorporarem as vantagens do JXTA em seus programas, utilizando-se a mesma interface de programação do soquetes TCP/IP da linguagem Java. Ele efetivamente esconde o JXTA provendo a ilusão que a rede Peer-to-Peer seja, de fato, uma rede padrão TCP/IP. Caso um membro da rede queira tornar-se servidor, ele simplesmente cria um soquete servidor com um nome de domínio que desejar e uma porta para os demais poderem encontrá-lo. Os clientes do ambiente colaborativo abrem conexões com os provedores de serviço através de soquetes Peer-to-Peer simples. Atrás da interface com o desenvolvedor, o P2PSockets utiliza as primitivas e os blocos fundamentais do JXTA para dar suporte ao projeto de reimplementação das classes básicas para comunicação remota da linguagem Java.

O P2PSockets utiliza internamente anúncios para publicar os recursos da rede

Peer-to-Peer (nós, grupos, serviços), estabelece e divulga os canais utilizados nas comunicações entre os nós, encapsula e troca mensagens no formato XML, entre outras tarefas atribuídas às aplicações JXTA típicas. Também são funções do P2PSockets gerenciar o acesso e utilização dos nós *rendezvous* (totalmente omitidos do usuário) e assegurar que quaisquer dois nós da rede possam se comunicar [KUR 04].

Percebe-se que os benefícios do emprego do P2PSockets são vários. Pesquisadores e desenvolvedores podem aproveitar seus conhecimentos relacionados aos soquetes TCP/IP para às redes Peer-to-Peer, sem a necessidade de conhecer em detalhes a especificação do projeto JXTA. A reescrita de aplicações elaboradas sob o modelo cliente-servidor para o Peer-to-Peer é rápida. Além disso, desenvolver sistemas baseados no P2PSockets e JXTA implica utilizar códigos construídos e validados por um conjunto extenso de cientistas[6], adotar projetos apoiados por grandes organizações como a Sun Microsystems e padronizar (mesma infraestrutura de comunicação) o modo de programação de sistemas Peer-to-Peer.

3.3 Balanço

Os projetos JXTA e P2PSockets representam o estado da arte no desenvolvimento de sistemas Peer-to-Peer. Este capítulo preocupou-se em retratar suas características e forma de funcionamento, de modo que o entendimento do protótipo desenvolvido durante a pesquisa, o qual absorve os conceitos destes dois projetos, seja facilitado.

O JXTA fornece uma infraestrutura robusta para a construção de aplicações P2P. Ele estabelece seis protocolos principais, cada um com uma função específica dentro do ambiente colaborativo. Ele também define os seguintes componentes básicos de um sistema Peer-to-Peer: nós, grupos, serviços, canais, anúncios e mensagens. Entre as principais vantagens do JXTA está a comunicação entre quaisquer dois elementos da rede P2P, independente de suas localizações ou projeto de suas redes locais.

O P2PSockets permite a escrita de aplicações P2P na linguagem Java mantendo-

[6]O projeto P2PSockets foi estabelecido no princípio de 2003 e possui 20 colaboradores ativos. O código-fonte do projeto é aberto e estável (passou pelas versões Alfa e Beta).

se a mesma interface de comunicação (soquetes) utilizada no processo de desenvolvimento de sistemas cliente-servidor. Ele esconde os detalhes do JXTA e possibilita que os usuários usufruam dos vários benefícios deste projeto, sem necessariamente conhecê-lo profundamente. As características do P2PSockets auxiliam para o crescimento das redes P2P, já que seu foco é facilitar o porte de aplicações elaboradas sob o paradigma padrão na Internet para o Peer-to-Peer.

Capítulo 4

Segurança Computacional

O tema central deste capítulo é a segurança de sistemas. Ele possui duas seções principais. A finalidade da seção 4.1 é apresentar os conceitos e definições fundamentais desta área, como as propriedades e mecanismos de segurança.

A seção 4.2 discute um dos fundamentos do processo de segurança - os sistemas de controle de acesso. Ela se interessa especificamente pelo processo de autorização e pela exposição dos três principais modelos de controle de acesso. Em ordem, são exibidos o controle de acesso discricionário (DAC), o controle de acesso obrigatório (MAC) e o controle de acesso baseado em papéis (RBAC). Esse último ganha uma atenção especial, pois insere-se grandemente nos objetivos desse trabalho.

4.1 Aspectos Gerais

A segurança de sistemas (computadores e informações) está entre as áreas da computação com maior proeminência, devido especialmente à importância dela no cotidiano das pessoas e negócios empresariais [STA 03]. A segurança trata da proteção dos ativos digitais armazenados em computadores e redes de processamento de dados [VEN 03].

Pode-se dizer que um computador é seguro se ele está livre de vulnerabilidades e preocupações a respeito de ameaças. A segurança computacional, neste sentido, é a dis-

ciplina que ajuda-nos a ficar despreocupados com os computadores, sendo assim possível reconhecer a palavra “seguro” como um atributo de um sistema ou objeto [LAN 01].

Para indicar um sistema como sendo seguro, ele deve manter três propriedades básicas: confidencialidade, integridade e disponibilidade. A confidencialidade é a propriedade de certas informações que não podem ser disponibilizadas ou divulgadas sem autorização. Garantir a confidencialidade de uma informação é permitir que apenas pessoas autorizadas possam acessá-la.

Analogamente, a integridade pode ser descrita como a condição na qual recursos são protegidos contra modificações sem prévia autorização. Um recurso é reconhecido como íntegro quando existe certeza que seu conteúdo é verdadeiro e original. A disponibilidade, por sua vez, é responsável por garantir que a informação estará acessível para os usuários legítimos quando estes requiserem. Segundo Landwehr [LAN 01], a autenticidade e o não-repúdio também são consideradas propriedades que precisam ser conquistadas para garantir a existência de um sistema seguro.

Para assegurar que os sistemas implantem as propriedades citadas anteriormente e sejam ditos seguros, existe a necessidade de adoção de mecanismos de segurança. Os mecanismos de segurança são os responsáveis efetivos pela garantia das propriedades e políticas de segurança. A Tabela 4.1, conforme escrito por Pernul [PER 95], agrega os mais notáveis mecanismos de proteção. Observe que Pernul trata a autenticação como um mecanismo de segurança e não com uma propriedade de sistemas seguros.

A política de segurança relaciona as propriedades e mecanismos de segurança a um domínio, além de definir o escopo e as características de cada serviço que se pretende proteger [UCH 01]. Ela determina regras que, quando seguidas corretamente, diminuem os riscos de incidentes de segurança à organização. Não existe como garantir a totalidade da segurança de um sistema; o que se busca é alcançar patamares admissíveis para o problema. Conforme Landwehr [LAN 01], um sistema ou organização sem uma política de segurança pode ser comparado com uma sociedade sem leis.

Outros termos relacionados à segurança são os riscos, ameaças e vulnerabilidades ocorridos em sistemas computacionais. Uma vulnerabilidade é um ponto suscetível ao ataque em um sistema. Ela pode ser um programa com falhas que executa de modo pri-

Mecanismo	Definição
Criptografia	Transforma dados em algo ininteligível para o inimigo, isto é, esconde o seu conteúdo semântico.
Autenticação	Verifica se uma entidade é quem afirma ser.
Autorização	Processo de determinar que tipo de atividades são permitidas.
Auditoria	Exame dos registros e das atividades do sistema para avaliar sobre sua confiabilidade.

Tabela 4.1: Principais mecanismos de segurança

vilegiado, uma senha fraca, regras mal configuradas no *firewall*, entre outros. Ameaças são intenções de danificar o sistema. Os riscos são a exposição de sistemas com falhas às vulnerabilidades. A política para minimizar os riscos é manter o sistema livre de vulnerabilidades.

Os sistemas de controle de acesso constituem um dos pontos principais desse trabalho. A próxima seção apresenta as relações existentes entre autenticação, autorização e controle de acesso, e as principais formas para garantir que apenas pessoas legítimas acessem recursos e informações em sistemas computacionais.

4.2 Sistemas de Controle de Acesso

Os sistemas de controle de acesso permitem determinar quais atividades ou operações são permitidas aos usuários legítimos. Sua principal função é definir o que cada entidade do sistema está autorizada a realizar [TRI 04]. Para efetivar seus objetivos, os sistemas de controle de acesso precisam garantir a identidade dos usuários e entidades envolvidas no processo. Nessa função eles fazem uso de mecanismos de autenticação, os quais atuam como suporte para a execução das demais atividades.

O processo de autenticação é responsável por provar ao sistema que o usuário é realmente quem diz ser, e não alguém se passando por ele. A autorização, elemento

chave do sistema de controle de acesso, determina o que alguém pode fazer. Portanto, a autenticação é uma condição prévia para a autorização.

Existem três formas principais de garantir a identidade dos usuários. A primeira é baseada em algo que o usuário conhece. Esse método é o mais comum, sendo que nele o usuário transmite ao sistema uma informação que apenas ele sabe (por exemplo, uma senha). No segundo método é utilizado algo que o usuário possui, como um cartão magnético, um *token* ou uma chave. O terceiro modo de autenticação verifica algo que o usuário é (característica física, como a impressão digital). Esse é o tipo menos comum entre os três citados, porém é o mais robusto. Nos dois primeiros modos de autenticação os problemas mais comuns são, respectivamente, a distribuição não autorizada da informação que se sabe (deixa de ser secreta) e a perda do objeto possuído.

Em um sistema de controle de acesso o termo sujeito representa toda a entidade que pode realizar operações sobre os objetos do modelo. Os sujeitos podem ser usuários, processos, entre outros. Um objeto identifica um recurso computacional cujo acesso é controlado. Um objeto pode ser definido como uma entidade que sofre as operações lançadas pelos sujeitos [MOT 03]. Em recursos do tipo arquivos e diretórios, as permissões mais comuns são leitura, escrita e execução.

A decisão de autorizar ou não o acesso a um objeto do sistema, muitas vezes, é função do monitor de referência. Conceitualmente, o monitor de referência é a porção de *hardware* e *software* de um sistema que é responsável por colocar em prática a política de proteção de ativos digitais adotada pela organização. Todas as tentativas de acesso a objetos realizadas pelos sujeitos são interceptadas e analisadas pelo monitor de referência. Logo após, ele concede ou nega o acesso ao objeto.

4.2.1 DAC – Controle de Acesso Discricionário

Os modelos de controle de acesso discricionário e obrigatório tiveram sua origem no meio militar. Em 1983, o departamento de defesa dos Estados Unidos (DoD) publicou os princípios desses dois modelos em um documento denominado "*Trusted Computer System Evaluation Criteria*" (TCSEC), também conhecido como "livro laranja".

O DAC baseia-se na idéia que os usuários ou sujeitos do sistema são proprietários de objetos e, portanto, têm controle total sobre quem deve ter acesso aos objetos. No DAC um usuário torna-se dono do objeto após criá-lo. Ele pode fornecer o direito de acesso ao recurso para outros usuários e retirá-lo quando entender necessário. Essas operações acontecem sem a intervenção do administrador do sistema.

O DAC restringe o acesso aos objetos do sistema baseado na identidade dos usuários, grupos ou de ambos. A maneira mais comum de implementar uma política de controle de acesso no modelo discricionário é através de listas de controle de acesso (ACL) localizadas junto aos objetos protegidos. Quando um sujeito tenta efetuar uma operação em um objeto, o monitor de referência verifica na lista de controle de acesso do objeto se existe alguma autorização que concede o privilégio. Existindo a autorização, o acesso é concedido e, caso contrário, ele é bloqueado.

Uma ACL representa uma maneira eficaz de implementar uma matriz de controle de acesso. A Tabela 4.2 exemplifica uma matriz de controle de acesso. Nela cada linha representa um sujeito do sistema e as colunas são utilizadas para designar os objetos. A intersecção de uma linha e uma coluna informa quais as operações que o sujeito pode realizar sob determinado objeto. Caso esta intersecção esteja vazia (sem valor), não existe nenhum tipo de acesso permitido sob o objeto para o sujeito em questão. O maior problema encontrado nesse tipo de construção é que para um sistema muito grande, a tabela torna-se extensa e parcialmente completa.

	Arquivo 1	**Arquivo 2**	**Arquivo 3**	**Processo 1**
João	Ler, Escrever	—	Escrever	—
Maria	—	Executar	—	Suspender
Fernando	—	Ler	Ler	—
Rodrigo	Ler	—	—	—

Tabela 4.2: Exemplo de uma matriz de controle de acesso

Quando utiliza-se listas de controle de acesso, as colunas da matriz de acesso tornam-se listas de usuários com capacidades de efetuar operações sob o objeto apontado

em cada linha. A principal vantagem dessa organização é a facilidade de verificar quais usuários possuem acesso a determinado objeto, assim como as operações que eles podem realizar sob esse objeto. Outra vantagem é que a lista de permissões para cada recurso não precisa ser muito longa; basta unir os usuários com características afins em grupos e utilizá-los no processo de autorização. A Tabela 4.3 expõe como são organizadas as permissões em uma ACL.

Objeto		
Arquivo 1	João: Ler, Escrever	Rodrigo: Ler
Arquivo 2	Maria: Executar	Fernando: Ler
Arquivo 3	João: Escrever	Fernando: Ler
Processo 1	Maria: Suspender	

Tabela 4.3: ACL – Lista de Controle de Acesso

Outro modo de representar uma matriz de controle de acesso é através do uso de *capabilities*. As *capabilities* são qualidades ou habilidades que determinado sujeito possui para realizar alguma tarefa. A Tabela 4.4 mostra o funcionamento desse esquema. Observa-se que cada usuário possui sua própria *capability*, representada através de uma lista que contém todos objetos e os direitos de acesso correspondente ao sujeito. A principal vantagem desse método é a facilidade de visualização e revogação de direitos atribuídos a um usuário. Porém, é difícil verificar todos os sujeitos que possuem acesso a um objeto em particular. É complicado revogar o acesso a determinado objeto para todos os usuários, pois seria necessário um exame completo de todas as *capabilities* do modelo.

O modelo de controle de acesso discricionário tornou-se muito popular por sua utilização em grande escala pelos sistemas operacionais. Os sistemas Unix e a série Windows NT, 2000 e XP utilizam o modelo DAC como seu modelo básico de controle de acesso [dAM 03]. O mecanismo de controle de acesso encontrado nos sistemas operacionais Unix é denominado “bits de proteção” (*protection bits*) [FER 03].

O esquema de bits de proteção é similar às ACLs; no entanto, ao invés de associar usuários e operações aos objetos, bits são associados com eles. Nesse modo de proteção os

Sujeito		
João	Arquivo 1: Ler, Escrever	Arquivo 3: Escrever
Maria	Arquivo 2: Executar	Processo 1: Suspender
Fernando	Arquivo 2: Ler	Arquivo 3: Ler
Rodrigo	Arquivo 1: Ler	

Tabela 4.4: Lista de *capabilities*

usuários são divididos em três categorias: proprietário do arquivo, grupo e outros (todos os demais). Para cada categoria são designados três bits, representando as operações de leitura, escrita e execução. Um arquivo que possui a permissão (`rwx`) (`r--`) (`---`) possibilita ao seu proprietário realizar qualquer operação sobre ele. Ao grupo apenas é permitido o acesso de leitura e nenhum acesso é fornecido aos demais usuários.

O modelo DAC possui algumas fraquezas importantes. A principal delas é que as políticas discricionárias não controlam a disseminação da informação [ROS 04]. Por exemplo, quando um usuário `A` permite que o usuário `B` tenha acesso de leitura ao arquivo, ninguém impede `B` de copiar o conteúdo desse arquivo para outro objeto, do qual possui total controle (ele é o proprietário). O usuário `B` pode agora fornecer o acesso a essa informação sem o prévio consentimento do usuário `A`.

4.2.2 MAC – Controle de Acesso Obrigatório

O modelo de controle de acesso obrigatório[1] obteve grande aceitação em ambientes militares. Ele preocupa-se sobretudo com a confidencialidade de informações sensíveis (preserva a leitura e a observação). Nesse modelo os usuários individuais não são considerados donos dos objetos e não possuem o direito de estabelecer permissões para cada um deles. O gerente do sistema é o único responsável por essa função.

No MAC os usuários e os objetos são classificados em níveis de segurança, ou seja, cada entidade possui um rótulo que informa a qual nível ela pertence. O rótulo atribuído a um objeto reflete a importância da informação mantida no objeto, isto é, o potencial de

[1] O MAC também é chamado de modelo de controle de acesso mandatório ou compulsório.

dano causado pela revelação ou alteração não autorizada da informação [SAN 94]. Em ambientes militares e hierárquicos os níveis de proteção mais comuns seguem a seguinte ordem: `ultra secreto` > `secreto` > `confidencial` > `não-classificado`. Diz-se que cada nível domina a si mesmo e todos os outros abaixo dele.

O controle de acesso formulado Bell e LaPadula (BLP) [BEL 76] para sistemas militares enquadra-se nos sistemas MAC. Ele afirma que os usuários somente podem ter acesso de leitura a um objeto se o seu nível[2] for maior ou igual à classificação do objeto. Conforme esse princípio, por exemplo, um soldado com acesso "confidencial" não pode ler um documento classificado como "secreto". Essa prática declara que a leitura de informações só funciona em um sentido.

Outro princípio importante encontrado no modelo de Bell e LaPadula é chamado de "*no write down*". Ele estabelece que um usuário de determinado *clearance* somente pode escrever em objetos que possuam classificação igual ou superior ao seu nível de habilitação. Esse conceito evita que um usuário com um nível de classificação alto copie um objeto do sistema e coloque nele uma classificação baixa. Portanto, ele auxilia para o não vazamento de informações sensíveis.

Quando um usuário autentica-se com o sistema e torna ativo o nível "secreto", seus programas e processos não possuem permissão para escrever informações no nível confidencial, mas podem escrever no próprio nível e em um nível superior, no caso o nível "ultra-secreto". Outra noção relevante nesse modelo de controle de acesso é que um usuário pode ter habilitação para atuar em vários níveis (por exemplo, "não-classificado" e "confidencial"). No entanto, apenas um nível pode estar ativo por vez, ou seja, no momento que o usuário apresenta-se para o sistema ele escolhe o nível que deseja acionar e este torna-se o nível ativo.

Os dois princípios (ou regras) definidos por Bell e LaPadula em seu modelo encontram-se abaixo. O símbolo ϕ significa o rótulo de segurança do sujeito ou objeto.

- *Simple-security property (no read up)*: um sujeito s pode ler o objeto o somente se $\phi(s) \geq \phi(o)$;

[2] O nível de classificação dos sujeitos muitas vezes é chamado de habilitação ou *clearance*.

- **-property (star property, no write down)*: um sujeito `s` tem permissão de escrita sob um objeto `o` somente se $\phi(s) \leq \phi(o)$.

O modelo de controle de acesso obrigatório é menos flexível que o controle de acesso discricionário. Como mencionado, ele obteve maior aprovação em ambientes militares. Contudo, com algumas adaptações, o MAC pode ser implantado em ambientes comerciais. Nesse caso, possíveis níveis de classificação poderiam ser `restrito` > `proprietário` > `sensível` > `público` [ROS 04].

4.2.3 RBAC – Controle de Acesso Baseado em Papéis

O RBAC é um modelo de controle de acesso baseado em papéis, no qual as permissões são associadas com os papéis e os usuários são classificados por funções. Este modelo distancia-se do habitual encontrado nos sistemas de proteção de informações, onde os usuários e seus direitos são intrinsecamente vinculados.

A Figura 4.1 exibe a estrutura principal do modelo RBAC, onde percebe-se que os usuários não são relacionados diretamente às permissões. Esse esquema auxilia na escrita de regras de controle de acesso, já que traz flexibilidade às relações existentes entre os usuários, papéis e permissões. Um exemplo dessa característica é observado quando um novo usuário é incluso em um sistema de computação e apenas lhe são designados seus papéis, ao invés de todos os seus potenciais direitos sob as aplicações e objetos envolvidos no conjunto.

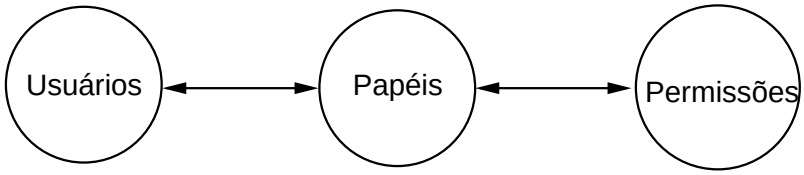

Figura 4.1: Relacionamentos entre usuários, papéis e permissões

A permissão é um conceito abstrato que se refere a vinculação (ou ligação) de uma operação a um objeto. A coleção de permissões designada a um papel confere o direito de executar tarefas, funções ou outras atividades relacionadas ao ambiente onde o RBAC está

inserido. A associação de um usuário com um papel lhe fornece a habilidade de explorar todos as permissões vinculadas a esse papel. A Figura 4.2 expande a representação do modelo RBAC através da inclusão dos elementos “operação” e “objeto”.

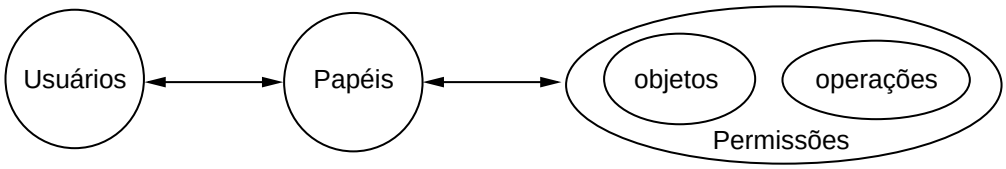

Figura 4.2: Elementos que compõem um modelo RBAC

O modelo RBAC afirma que um usuário pode possuir mais de um papel, assim como um papel pode estar associado a várias permissões. Quando um usuário possui determinado papel, ele pode realizar todas as atribuições daquele papel. Como um usuário executa somente o que lhe foi expressamente permitido, diz-se que o RBAC implementa o princípio de segurança chamado “menor privilégio”.

O modelo RBAC facilita a administração da política de controle de acesso através da utilização do conceito de papéis. Quando um usuário muda de função, basta alterar os papéis atribuídos a ele. Da mesma forma, os papéis também podem receber novas permissões à medida que novas aplicações ou funcionalidades são adicionadas ao sistema. O modelo RBAC possibilita que permissões sejam revogadas facilmente sempre que necessário.

Estudos realizados pelo NIST[3] em ambientes que utilizam o RBAC concluíram que a freqüência de alteração de usuários é muito maior se comparada com as mudanças nos papéis. O papel é mais estável porque as atividades e funções da organização, em geral, mudam menos do que o conjunto de usuários ou permissões. Esse fato reforça os benefícios do RBAC, pois mostra que se a cada alteração de usuário o administrador tivesse que preencher todas as suas permissões, o tempo para efetuar a manutenção no sistema de controle de acesso seria elevado.

Como mencionado, o modelo RBAC utiliza papéis como elementos centrais no processo de autorização. O conceito de papéis não deve ser confundido com o conceito

[3]*National Institute of Standards and Technology*.

de grupos. Os grupos, na terminologia de sistemas de controle de acesso, são coleções de usuários, ao invés de um conjunto de permissões. Um papel é tanto uma coleção de usuários de um lado, e uma coleção de permissões no outro [dAM 03]. Outra diferença entre esses dois termos é que os papéis podem ser hierárquicos.

O RBAC prevê a possibilidade da existência de hierarquia de papéis. Desta maneira, um usuário que obtém um papel Y obtém também todos os demais papéis abaixo de Y na hierarquia. Essa particularidade agrega complexidade ao modelo, mas deixa-o mais perto da compreensão do mundo real em sua totalidade. Outra característica possível é a adoção de restrições no relacionamento entre os papéis onde, por exemplo, dois determinados papéis não podem ser designados juntos ao mesmo usuário.

Outra característica do RBAC é que ele permite a "separação de responsabilidades"[4] (SoD - *Separation of Duties*), cujo objetivo é reduzir as chances de fraude ou dano acidental decorrente da demasiada concentração de poder em uma única pessoa [MOT 03]. Por esse princípio, um usuário não pode subverter a segurança do sistema ao exercer dois papéis conflitantes ao mesmo tempo. A separação de responsabilidades é um paradigma largamente usado nos ambientes corporativos; fato que colaborou para a aceitação do RBAC nesses meios.

O NIST iniciou em 2000 um esforço para estabelecer um padrão internacional para o RBAC. Nessa ocasião, foi publicado uma proposta para o modelo no evento ACM RBAC Workshop (atualmente chamado de Sacmat[5]). Essa proposta de padrão seguiu a estrutura do RBAC96 [SAN 96] e incorporou estudos oriundos do meio científico e comercial. O objetivo desse documento foi normatizar o escopo, os conceitos e as terminologias envolvidas no RBAC. Em 2002 a proposta foi submetida ao processo internacional de padrões e várias organizações, desde então, vêm desenvolvendo sistemas de segurança em conformidade com o padrão RBAC.

[4] Também chamada de separação de deveres.

[5] *Symposium on Access Control Models and Technologies*

4.2.3.1 Modelo RBAC Definido pelo NIST

O RBAC é um modelo e não um mecanismo de controle de acesso. Ele pode ser implementado tanto em sistemas simples quanto em sistemas mais complexos e sofisticados. Em todos os casos, necessariamente, há a presença da entidade "papel" para intermediar a atribuição de direitos aos usuários.

O modelo RBAC proposto pelo NIST é decomposto em quatro outros modelos. Seus nomes são `RBAC0`, `RBAC1`, `RBAC2` e `RBAC3`. O `RBAC0` representa a especificação básica do controle de acesso baseado em papéis. Os modelos `RBAC1` e `RBAC2` englobam o `RBAC0` e adicionam, respectivamente, o suporte a hierarquia de papéis e restrições (como o SoD) às funcionalidades já existentes no modelo básico. Por sua vez, o `RBAC3` envolve os três modelos anteriores e constitui, assim, o mais completo entre eles. A Figura 4.3 exibe os quatro modelos mencionados.

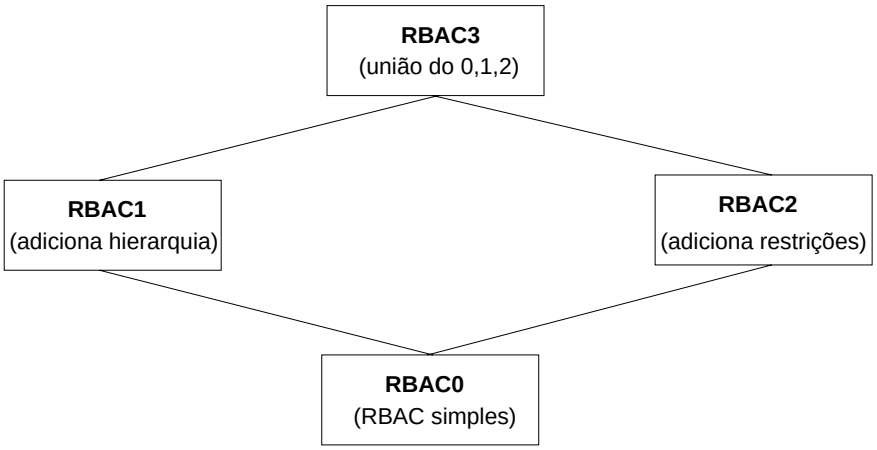

Figura 4.3: Estrutura dos modelos RBAC

RBAC Principal ou RBAC0

Esse modelo possui as características mínimas do RBAC. No RBAC papéis são designados aos usuários. Esse vínculo pode representar as competências, autoridades ou responsabilidades de um usuário. Quando um usuário é associado a um papel, diz-se que ele está autorizado para executar as atribuições daquele papel. O modelo `RBAC0` é ilustrado na Figura 4.4.

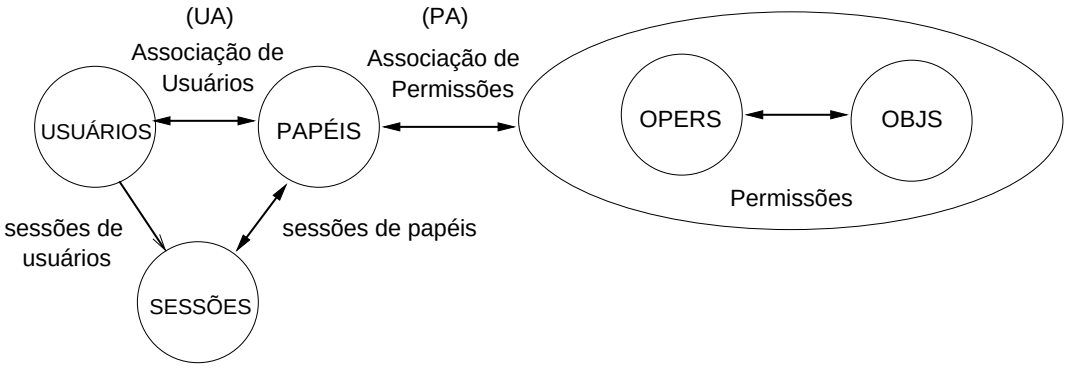

Figura 4.4: Estrutura do Modelo RBAC0

As siglas UA e PA da Figura 4.4 representam os vínculos existentes entre os usuários e os papéis e entre os papéis e as permissões. Um usuário pode estar associado a vários papéis, assim como um funcionário de uma organização pode exercer mais de um cargo na mesma empresa. Algumas implementações do RBAC, contudo, determinam que o usuário utilize um papel por vez no sistema. Já outras construções permitem o acúmulo de direitos de acesso. Nesse caso, o usuário pode exercer mais de papel no sistema ao mesmo tempo.

Segundo Lincoln [dAM 03], a permissão é a aprovação de um modo de acesso sob algum recurso do sistema. As permissões são sempre positivas e, como relacionado anteriormente, elas conferem alguma habilidade ao usuário. O `RBAC0` permite que cada implementação do RBAC utilize seu próprio método para gerir as permissões do sistema. Em um SGBD (Sistema de Gerência de Banco de Dados) as permissões podem ser “insert”, “update”, “delete”, enquanto em um sistema operacional os direitos mais comuns são “leitura”, “escrita” e “execução”.

RBAC Hierárquico ou RBAC1

A Figura 4.5 mostra a estrutura dos elementos no RBAC hierárquico. A principal diferença entre o `RBAC1` e o `RBAC0` é a presença da hierarquia de papéis. A motivação para a criação do modelo `RBAC1` é a observação que papéis individuais em uma organização muitas vezes possuem funções sobrepostas, isto é, usuários que perten-

cem a papéis diferentes são autorizados para executar permissões em comum.

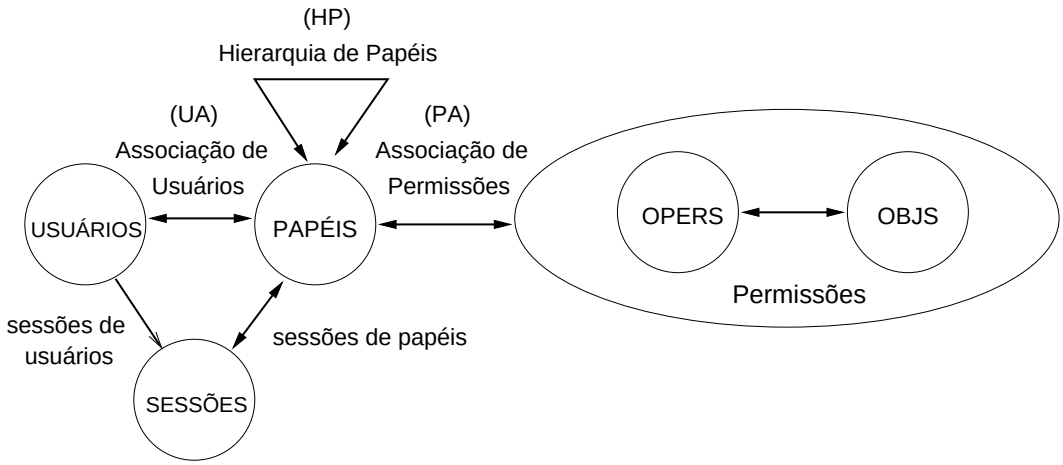

Figura 4.5: Estrutura do Modelo RBAC1

O uso de hierarquia de papéis facilita a administração do modelo RBAC, pois as permissões atribuídas a um papel-filho não incluem aquelas vinculadas ao papel-pai. A Figura 4.6 exemplifica o uso de hierarquia de papéis. Nesse exemplo os papéis cardiologista e pediatra herdam os papéis médico e residente. Todo usuário associado ao papel cardiologista pode realizar as funções atribuídas a esse gênero de profissional e também aquelas estabelecidas para os papéis médico e residente.

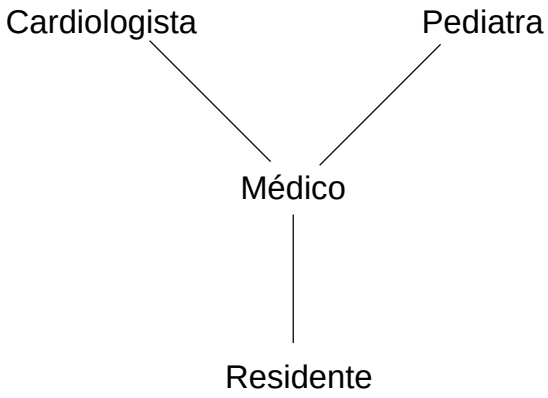

Figura 4.6: Exemplo de hierarquia de papéis [FER 03]

RBAC com Suporte a Restrições ou RBAC2

As restrições[6] possibilitam que o RBAC se encaixe em ambientes diversos e seja mais rico em detalhes. As restrições são aplicadas às relações e funções do modelo; elas retornam um valor “aceito” ou “bloqueado”. Elas são utilizadas para expressar a separação de responsabilidades (SoD), onde dois papéis são mutuamente excludentes. A Figura 4.7 exibe a estrutura do RBAC2. As siglas SSD e DSD designam a separação estática e dinâmica de responsabilidades.

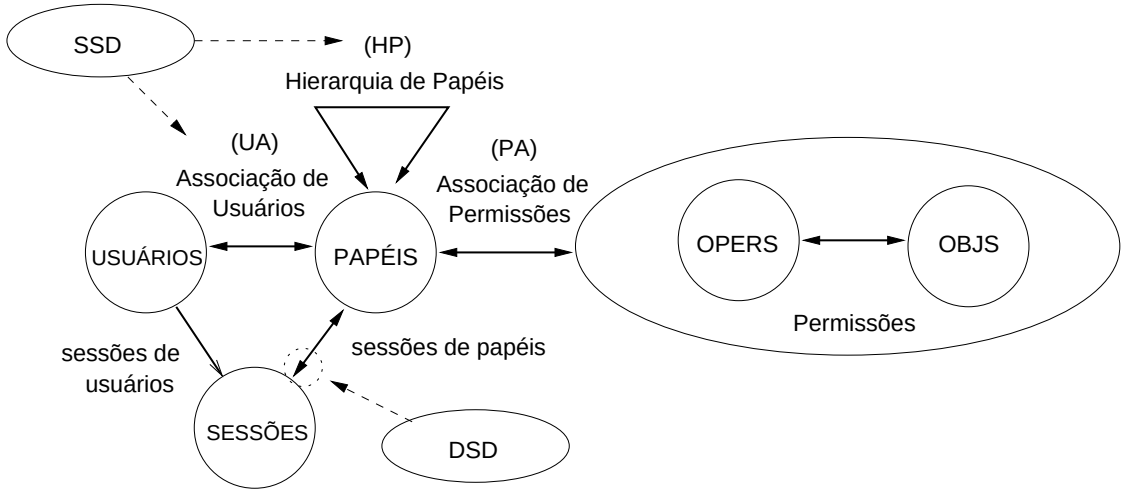

Figura 4.7: Estrutura do Modelo RBAC2

As restrições podem ser ativadas em diversos pontos do modelo RBAC, como no relacionamento entre usuários e papéis (UA) e entre papéis e permissões (PA). Outros tipos de restrições são aquelas impostas às sessões estabelecidas pelos usuários.

Com a utilização de restrições pode-se determinar, por exemplo, o número máximo de papéis que podem ser associados a um usuário, o horário que determinado usuário tem permissão para acessar o sistema, papéis mutuamente exclusivos, entre outros. Percebe-se a riqueza de observações que podem ser praticadas no RBAC2.

O RBAC2 estabelece dois tipos de separação de responsabilidades: a estática e a dinâmica. A separação de responsabilidades estática (SSD) evita o conflito de interesse dentro de um modelo de controle de acesso. Um exemplo comum é evitar que um usuário

[6]Esse documento traduz o termo *constraint* para “restrição”.

seja associado ao papel “Caixa” e “Fiscal de Caixa” durante a configuração do modelo de autorização. Já a separação de responsabilidades dinâmica possui uma abordagem diferente. Ela age sobre o estabelecimento de sessões dentro do sistema. Comparando com o exemplo anterior, nesse caso o usuário poderia ter associado a si os papéis “Caixa” e “Fiscal de Caixa”. O empecilho é que ele não pode exercer os dois papéis ao mesmo tempo. Sempre que for autenticar-se com o sistema, o usuário declara qual dos dois papéis com restrição dinâmica deseja ativar.

4.3 Balanço

Este capítulo apresentou os principais assuntos relacionados à segurança e ao controle de acesso em sistemas computacionais. Em geral, a segurança computacional pode ser definida como a prevenção e detecção de ações não autorizadas em um sistema ou ambiente. A segurança prevê o cumprimento de três propriedades fundamentais que são a confidencialidade, a integridade e a disponibilidade.

O controle de acesso é um dos blocos fundamentais do processo de segurança. É ele quem determina o que cada entidade do sistema está autorizada a realizar. Os três principais modelos de controle de acesso são o discricionário, o obrigatório e o baseado em papéis. Este último foi o modelo de controle de acesso escolhido para compor a arquitetura P2P-Role, a qual representa o ponto principal desta dissertação de mestrado.

O modelo RBAC associa indiretamente os usuários às suas permissões de acesso. Ele utiliza o conceito de papéis para intermediar o processo de autorização e consegue, desta forma, vários benefícios como o facilidade na gerência das políticas de segurança das organizações que adotam este modelo. O capítulo atual tratou da segurança computacional, porém não faz nenhuma relação deste tópico com as redes e sistemas Peer-to-Peer. O capítulo 5 apresenta os principais estudos realizados em segurança de redes P2P e os trabalhos relacionados ao tema de pesquisa escolhido nesta dissertação.

Capítulo 5

Estado da Arte em Segurança de Redes Peer-to-Peer

Os sistemas Peer-to-Peer foram originalmente desenvolvidos sem maiores preocupações com a segurança. O foco das aplicações era criar um ambiente com uma funcionalidade específica (por exemplo, compartilhamento de músicas) e uma infraestrutura que fosse estável para suportar a comunicação entre os elementos da rede. A evolução das redes P2P foi acompanhada pelo aprimoramento das técnicas de segurança inseridas nesses sistemas. Tornar uma rede P2P segura significa garantir principalmente a confiabilidade das comunicações, a execução correta dos protocolos envolvidos e o convívio harmonioso dos usuários da comunidade virtual.

A seção 5.1 apresenta os temas mais relevantes e atuais na área de segurança de sistemas Peer-to-Peer e descreve as pesquisas que abordam esses assuntos. A seção 5.2 possui grande importância, pois especifica os trabalhos científicos que associam os sistemas P2P e os modelos de autorização. Ela estabelece as pesquisas que tratam do mesmo tema escolhido por essa dissertação – o controle de acesso em redes Peer-to-Peer. O capítulo 5 encerra na seção de Balanço com a reunião dos tópicos fundamentais discutidos.

5.1 Proteção de Sistemas Peer-to-Peer

A computação em redes colaborativas exige que muitos paradigmas existentes nas aplicações cliente-servidor atuais sejam repensadas a fim de serem adaptados a esse novo modelo. Esse é o caso da segurança computacional, a qual precisa manter as propriedades de confidencialidade, disponibilidade, integridade [LAN 01] também nas redes Peer-to-Peer. O trabalho de Barkai [BAR 01] relaciona os seguintes requisitos de segurança para as redes P2P:

- o nó P2P precisa ter certeza que o outro ponto da conexão é quem diz ser;
- o nó P2P deve conhecer quando e como os demais elementos da rede irão acessar os seus recursos;
- os integrantes do sistema descentralizado precisam ter a certeza que as suas mensagens não serão lidas ou modificadas durante o processo de roteamento.

Esta seção apresenta alguns trabalhos científicos que exploram a segurança em redes Peer-to-Peer. São expostos trabalhos que examinam os ataques que as redes P2P podem sofrer, os algoritmos de reputação e micropagamentos (usados para proteger as redes P2P de atividades anti-sociais), a identificação dos elementos da rede, entre outros. Como relacionado, o tópico controle de acesso em redes Peer-to-Peer e os trabalhos relacionados nessa área são descritos na seção 5.2. Esse assunto possui maior destaque (seção própria), pois representa o tema principal analisado nesse documento.

Além dos requisitos de segurança, os sistemas P2P seguros precisam resolver os problemas de comportamento que podem surgir dentro da comunidade formada pelos elementos da rede. As redes P2P requerem regras que regulam o bom funcionamento do sistema (garantem o convívio social) e evitam atos como a modificação imprópria do protocolo P2P em benefício próprio ou a existência de elementos caronas[1] no ambiente colaborativo. Esses elementos não disponibilizam recursos para a rede Peer-to-Peer, agem apenas como sanguessugas de recursos e auxiliam para a queda de desempenho e existência de congestionamento na topologia de rede.

[1] Os nós caronas também são chamados de *free riders* e *free loaders* [MIL 02].

Os pioneiros no estudo dos efeitos dos nós caronas foram Adar e Huberman [ADA 00]. Eles descobriram em suas pesquisas que quase 70% dos usuários não compartilham recursos em uma rede P2P do tipo Gnutella e aproximadamente 50% de todas as respostas são retornadas por 1% dos hospedeiros compartilhadores. Assim, eles confirmaram a necessidade de haver mecanismos que incentivem a colaboração e a ajuda mútua nas redes Peer-to-Peer.

Desde então, vários autores têm investigado técnicas que estimulam os usuários a compartilhar recursos na rede P2P. O trabalho de [GOL 01] classifica os métodos de incentivo à colaboração em dois grupos. O primeiro emprega técnicas de micropagamento e nele os nós ganham algum valor ou moeda virtual "vendendo" seus recursos para a comunidade e os gastam quando consomem recursos. O segundo grupo engloba as técnicas que adotam a recompensa como fundamento principal. Nessa classificação os nós que possuem bons comportamentos são recompensados com vantagens, como o acesso a enlaces mais velozes na topologia de rede. O maior problema existente no primeiro grupo de técnicas é quantificar com clareza o custo (valor) de cada recurso do ambiente [PAP 04]. Situação semelhante ocorre com os métodos de recompensa, pois é preciso identificar quais atitudes são positivas para a rede e qual a gratificação ideal para cada uma delas.

O middleware desenvolvido por Strulo [STR 04] reúne as vantagens das duas classificações anteriores. Ele divide os participantes da rede P2P em grupos e neles são definidas regras de comportamento a serem seguidas por todos os seus integrantes. O modelo Escambo [RIG 04a] é outro trabalho que objetiva minimizar o número de nós caronas de um sistema P2P. Ele possui algumas características semelhantes ao trabalho de Strulo (conceitos de regras e políticas de comportamento) e sua premissa básica é a seguinte: "quem deseja informações da rede deve necessariamente também disponibilizar recursos". As técnicas de controle de comportamento e incentivo à colaboração do Escambo são as selecionadas para serem agregadas à arquitetura de controle de acesso P2P-Role desenvolvida nesta dissertação. O funcionamento do modelo Escambo é descrito no capítulo 6, junto com a contribuição principal deste trabalho.

O trabalho de Karakaya [KAR 04] pesquisa como é possível reduzir o número de nós caronas em uma rede P2P do tipo Gnutella. Em [KAR 04] cada nó da rede realiza

uma monitoração constante de seus vizinhos, a fim de detectar se eles estão agindo de maneira correta ou não. Verifica-se, para cada vizinho, o número de pedidos que ele faz, a quantidade de respostas positivas que ele emite e se seu comportamento é seguro (segue a especificação do protocolo Gnutella). Baseado na análise destas informações, cada participante pode optar por aplicar uma punição aos seus vizinhos. Entre as punições estão as seguintes: decrementar o campo "tempo de vida" (TTL) da mensagem de requisição do vizinho em um número maior que um (1) para evitar que seu pedido se propague em profundidade na rede; ignorar as requisições provenientes dos vizinhos classificados como nós caronas; quebrar (desfazer) a conexão com o vizinho malfeitor – quanto menor for a quantidade de vizinhos (conexões) de um nó, menores são os resultados de pesquisa que ele obtém da rede.

A segurança de projetos Peer-to-Peer puros, na maioria da vezes, é mais complexa de se atingir que a proteção de sistemas desenvolvidos no modelo cliente-servidor. Fatores como heterogeneidade dos elementos da rede e a inexistência de um ponto central contribuem para essa situação. Conforme Gupta [GUP 03], as redes Peer-to-Peer distribuem a responsabilidade pela segurança para todos os seus elementos, isto é, tornar a rede segura deve ser um esforço comunitário.

Outra questão relacionada às redes Peer-to-Peer e de grande importância aos projetistas de aplicações P2P seguras é a seguinte: os elementos da rede P2P são representados por usuários com pouca experiência em segurança de sistemas; muitos deles desconhecem totalmente o assunto. Portanto, a proteção das aplicações deve ser transparente e de fácil configuração. Diferentemente, na arquitetura cliente-servidor os servidores geralmente são localizados em centros de computação especializados e administrados por pessoal preocupado com as práticas de segurança.

Gupta [GUP 03] descreve os cenários de ataques que podem ser desenvolvidos contra as redes Peer-to-Peer puras. Ele apresenta primeiro os ataques aos sistemas P2P não estruturados e, logo em seguida, aqueles orientados às redes P2P estruturadas. Os ataques conhecidos como auto-replicação, "homem no meio" (*man in the middle*) e negação de serviço destinam-se principalmente às redes P2P não estruturadas.

O ataque por auto-replicação tira proveito dos sistemas P2P que usam identificadores não persistentes para representar os seus elementos. Nesse ataque, um nó malicioso age de forma errada na rede (por exemplo, corrompe o protocolo de reputação ou envia vírus escondidos em recursos que disponibiliza na rede) e quando os demais integrantes do sistema percebem o malfeitor e tentam retirá-lo do ambiente, ele apenas troca de identificador e volta à rede sem ser reconhecido. No ataque “homem no meio” o atacante posiciona-se entre os nós que se comunicam e, sem que nenhum deles perceba, ele captura as mensagens de conexão, modifica-as e entrega-as como legítimas a outra ponta do canal de comunicação.

O ataque de negação de serviço contra as redes Peer-to-Peer não estruturadas é bem simples. Nele, os nós maliciosos geram consultas na rede com o campo TTL muito alto. Essas mensagens são distribuídas segundo o algoritmo de inundação e permanecem na rede por um tempo elevado, congestionando-a. Esse cenário, conseqüentemente, deixa o sistema indisponível para os usuários legítimos.

O ataque às redes P2P estruturadas visam desestabilizar o algoritmo de roteamento utilizado (DHT). Nós maliciosos podem atacar esses sistemas realizando o roteamento de forma irregular não enviando a requisição para o vizinho que possui o identificador mais próximo àquele encontrado na solicitação ou simplesmente não repassando o pedido pelo recurso a frente. Ele retém a mensagem e impossibilita a recuperação de recursos.

No ataque de armazenamento um nó da rede Peer-to-Peer, mesmo que possua o recurso procurado, responde às requisições de maneira negativa. Outra idéia utilizada para impedir o uso correto da rede P2P é a inserção e, logo após, remoção de vários nós da rede continuamente. A cada operação dessas a topologia do sistema sofre adaptações. Portanto, quando o processo é realizado em alta escala pode-se ter um colapso do ambiente Peer-to-Peer.

O trabalho de Wallach [WAL 02] também relaciona as formas de inutilizar um sistema Peer-to-Peer. Porém sua abordagem é mais detalhada e direcionada às redes P2P que empregam o roteamento de recursos com DHT. Um dos problemas citados é a atribuição segura de identificadores aos elementos da rede. Um atacante poderia reunir vários nós maliciosos e adquirir de forma indevida identificadores que lhe permitissem ficar res-

ponsável por um determinado conjunto de chaves (recursos) que lhe interessam. Logo após, ele poderia usar o ataque de armazenamento para impedir que outros nós da rede recuperem essa informação.

Wallach [WAL 02] explica os métodos utilizados pelos sistemas P2P para a distribuição de identificadores aos elementos que se unem à rede. O mais comum é a utilização de uma autoridade central que emite certificados aos elementos, atribuindo a eles um identificador aleatório. Ao invés de associar uma chave pública com um endereço de e-mail, nesse contexto é associada uma chave pública a um identificador. A autoridade central é consultada somente quando os nós entram na rede. Assim, não existe impacto na escalabilidade do sistema.

Os sistemas P2P podem ser usados para reforçar a segurança da infraestrutura de rede de organizações. O trabalho [VLA 04] desenvolve uma arquitetura P2P chamada *NetBiotic*, na qual seus nós funcionam como agentes coletores de informações de segurança na rede. Os nós desta arquitetura comunicam-se com *firewalls*, detectores de intrusão, programas anti-vírus e também trocam mensagens entre si. O objetivo principal é detectar e remediar incidentes de segurança – vírus, worms, cavalos de tróia e outros – de maneira rápida e eficiente.

A reputação nos sistemas Peer-to-Peer informa quais participantes da rede são honestos ou bom servidores de recursos. A reputação dos nós é adquirida através de experiências dos próprios membros da rede e nas trocas de informações de reputação entre pares que confiam um no outro. O estudo produzido por Wang e Vassileva [WAN 03] apresenta um modelo de reputação e de confiança para redes Peer-to-Peer. Nele, cada elemento da rede possui duas tabelas de informações. Na primeira tabela são armazenadas as experiências que o nó obteve quando realizou o *download* de recursos dos demais integrantes da rede. Essa experiência representa o quanto o nó cliente está satisfeito com os recursos que recebeu e pode ser dividida em três categorias: velocidade do *download*, qualidade do arquivo e tipo de arquivo.

A partir dessa tabela o nó conhece os membros da rede que possuem mais possibilidades de lhe trazer bons recursos. Ainda nesse modelo, quando um nó necessita do valor de uma reputação, ele difunde o pedido (especifica a categoria) para os seus conhecidos

e aguarda as respostas. Logo após, ele ordena as respostas baseado nos elementos que ele mais confia para recuperar recursos (ele utiliza as suas experiências antigas para essa função). Nesse momento, o nó pode abrir uma conexão com algum elemento – de preferência com aquele localizado no topo da lista – ou questionar a rede sobre a reputação de algum outro integrante do ambiente. A segunda tabela de informações que os nós armazenam relaciona o quanto cada nó confia nas respostas de reputação enviadas pelos demais.

O pedido pela reputação de um nó é enviado apenas aos elementos mais confiáveis, do ponto de vista do membro que lança o pedido de reputação. Como a requisição de valores de reputação é enviada a um sub-conjunto de elementos, o número de mensagens provenientes desse processo não baixa o desempenho do sistema Peer-to-Peer. A segunda tabela contém os nós que enviam recomendações ao requisitor e o quanto este confia nas informações passadas por cada um deles. Portanto, a seguinte situação pode acontecer: o nó A confia que o elemento B possui bons recursos e boas características (por exemplo, largura de banda), porém não acredita nas suas recomendações sobre outros nós.

Para finalizar o modelo de Wang e Vassileva [WAN 03], sempre que uma interação com outro elemento da rede for positiva, o nó deve aumentar o índice de confiança sobre esse elemento. Usa-se para isso a primeira tabela. Se a decisão de interagir com determinado elemento proveio de uma recomendação, também é atualizada a confiança no membro da rede que passou a recomendação produtiva[2]. A segunda tabela é aplicada nesse caso. Os valores das duas tabelas também são alterados quando o *download* do recurso não satisfaz a origem ou quando as recomendações recebidas mostram-se não verdadeiras.

A pesquisa realizada por Marti [MAR 03] associa a reputação e a anonimidade. Ela informa que a identidade de um nó deve ser contínua na rede, já que o mecanismo de identificação dos nós é quem oferece suporte ao sistema de reputação no ambiente Peer-to-Peer. Com relação as formas de identificação dos elementos, observa-se no artigo a preferência pelo uso de certificados digitais para essa função em detrimento do endereço

[2]Diz-se que uma recomendação é produtiva quando quem a recebe confirma (*download* com sucesso) a indicação existente nela.

IP. O documento também informa as diversas políticas que um elemento pode utilizar para escolher qual a melhor resposta de reputação, dentre aquelas que ele recebe a cada pedido por esse tipo de informação na rede. Nesse sentido, ele cita dois métodos: o *Random Selection* (aleatório) e o *Select Best* (avaliar a melhor). Por fim, [MAR 03] reforça que o objetivo da reputação é reduzir o número de recursos que o usuário precisa examinar antes de encontrar uma informação que lhe satisfaz.

Outros dois trabalhos na área de reputação para redes Peer-to-Peer são [COR 02] e [DAM 02]. Eles descrevem o protocolo de reputação *XRep*, o qual inova em alguns paradigmas. No XRep cada nó recolhe índices de reputação para outros nós da rede Peer-to-Peer e também para cada recurso que entra em contato. Os nós podem requisitar aos demais membros da rede tanto informações sobre outros participantes, como informações sobre um recurso específico (se ele é autêntico ou não). A pesquisa da reputação de recursos auxilia para que um nó não tenha surpresas quando for abrir o recurso que buscou da rede. Um exemplo de situação que o *XRep* ajuda a evitar é o seguinte: um usuário realiza o *download* de uma música da rede Peer-to-Peer e, ao escutar, observa uma propaganda de um produto comercial.

O micropagamento introduz o conceito de pagamento pelo acesso a um recurso ou pedido de atividade. Esse pagamento pode ser de dois tipos: *(i)* aquele em que o nó servidor não recebe nenhum valor e o cliente paga-o geralmente com trabalho (por exemplo, para acessar um recurso o nó precisa realizar o cálculo de uma tarefa computacionalmente complexa); *(ii)* o cliente utiliza algum sistema de pagamento, como o dinheiro virtual, para pagar o detentor dos recursos, o qual tem acesso ao montante recebido. A utilização de micropagamentos auxilia a rede na proteção contra ataques de negação de serviço e no controle do congestionamento, pois sempre que um recurso for requisitado, algum pagamento deve ser realizado. Para um nó malicioso executar centenas ou milhares de micropagamentos pode tornar-se caro – se o pagamento for em moeda – ou computacionalmente inviável caso o pagamento for com ciclos de CPU. As pesquisas relacionadas aos micropagamentos em ambientes Peer-to-Peer são recentes, sendo um dos pioneiros no tema o trabalho de Garcia-Molina [GM 03].

A segurança em redes P2P também é o tema principal em [GAO 03]. Este artigo apresenta as características das redes P2P e afirma, como em outros trabalhos, que implantar mecanismos de proteção nesses ambientes é mais complicado se comparado aos modelos centralizados, principalmente devido à dificuldade de utilização de uma gerência de segurança uniforme para todo sistema Peer-to-Peer. São exibidos alguns métodos usados para estabelecer um canal seguro entre os pares de elementos. Entre eles estão o uso do HTTPS, SSL e TLS. A pesquisa declara que para a identificação dos participantes da rede pode-se utilizar certificados digitais em conjunto com uma infraestrutura de chaves públicas (PKI) robusta. Com relação a essa questão, o autor informa que muitas vezes é inviável a adoção de apenas uma autoridade certificadora (CA) para contemplar toda a comunidade Peer-to-Peer. Nesse caso, pode-se separar os nós em sub-grupos, cada um com a sua CA própria. Assim, a implantação de uma hierarquia de certificação será necessária.

A utilização de uma infraestrutura de chaves públicas em protótipos de segurança para redes P2P é uma tendência verificada em trabalhos como [SIN 03]. Ele utiliza o modelo de criptografia assimétrica (chave pública e chave privada) para estabelecer um sistema de reputação eficiente e anônimo. Nesse modelo existe um nó chamado *bootstrap* que intermedia a entrada de novos nós na rede. Quando um novo nó ingressa à rede, o *bootstrap* seleciona entre os participantes do sistema quais serão os responsáveis por armazenar o índice de reputação do nó que entra no ambiente. São chamados de elementos THA aqueles que guardam informações de reputação. A troca de mensagens na rede é totalmente cifrada e prioriza a anonimidade nas relações de requisição e resposta de reputações.

A adoção de técnicas de micropagamento e de reputação podem evitar a existência de vários ataques às redes Peer-to-Peer. A partir do momento que há um micropagamento para adquirir um identificador na rede, lançar consultas no sistema ou recuperar um recurso do ambiente, fica mais difícil para os os nós maliciosos realizarem tarefas como entrar e sair da rede continuamente na tentativa de tirá-la de operação, enganar o sistema de reputação ou agir como sanguessuga de recursos do ambiente. A reputação, por sua vez, possibilita o isolamento de nós maliciosos, dificulta a transmissão de vírus e mensagens indesejadas na rede (com o tempo esses recursos teriam baixa reputação e seriam

esquecidos) e favorece aqueles elementos que colaboram para a criação de um sistema Peer-to-Peer harmonioso.

5.2 Controle de Acesso em Redes Peer-to-Peer

Várias pesquisas definem a necessidade de agregar os mecanismos de controle de acesso às redes P2P [FEN 02, DAS 03, BER 04]. O estudo de Daswani e Garcia-Molina [DAS 03] (*Stanford Peers Group*) é um dos trabalhos que relaciona os requisitos de segurança para as redes Peer-to-Peer. Ele determina os principais problemas de segurança que ainda precisam ser aprofundados nas redes colaborativas P2P. Os temas citados são: garantia de disponibilidade da rede P2P, autenticidade de recursos (verificar se o recurso é mesmo o que diz ser), anonimidade e controle de acesso. Ele também confirma a tendência das aplicações Peer-to-Peer em utilizarem mecanismos de segurança nas suas comunicações. Isso acontece pois as aplicações P2P expandiram suas funcionalidades para além do compartilhamento de arquivos.

Os pesquisadores Fenkam e Kirda [FEN 02] afirmam que existem algumas características indispensáveis em uma arquitetura de controle de acesso voltada às redes Peer-to-Peer. Esse estudo apresenta o modelo DUMAS (*Dynamic User Management and Access Control*), utilizado para gerência de permissões em ambientes P2P com dispositivos móveis, e informa as seguintes propriedades dos sistemas de controle de acesso: descentralização do controle, suporte a vários protocolos de autorização, interface para administração da política de autorização em cada participante da rede e escalabilidade.

O trabalho de Berket [BER 04] relaciona como a infraestrutura de chaves públicas pode ser utilizada para reforçar a segurança das redes P2P. Em especial, Berket se preocupa com a autenticação e com a autorização. Ele utiliza certificados X.509 para solucionar os requisitos de autenticação. Já a autorização é tratada de dois modos. No primeiro cada nó tem uma política de configuração de canal[3], a qual informa as características mínimas (por exemplo, tipo e criptografia, identificação do grupo) que outros participantes devem preencher quando abrirem uma conexão consigo. Se os requisitos forem

[3] *Channel Configuration Policy*

satisfeitos, o canal é estabelecido e o acesso liberado. Berket também utiliza o conceito de certificados de *capabilities*. Os certificados atribuídos aos nós da rede possuem, junto com o conjunto de informações habituais (chave pública, dados pessoais, etc), atributos que informam quais ações eles estão autorizados a realizar na rede. A pesquisa [BER 04] possui bons resultados, porém o uso de certificados de *capabilities* dificulta a revogação de uma ação específica no sistema, pois seria necessário a investigação de todos os certificados existentes.

A pesquisa realizada por Park e Hwang [PAR 03] é o principal trabalho relacionado ao tema tratado nesse documento. Ela apresenta uma arquitetura que explora o tema controle de acesso em redes Peer-to-Peer. Esses pesquisadores projetaram um middleware que funciona como elemento intermediário entre os participantes da rede Peer-to-Peer. Dessa forma a arquitetura consegue prover um ambiente P2P controlado. Os componentes que formam essa arquitetura são: o Gateway de Comunicação, o Identificador de Serviços, o Gerente, o Gerador de XML e a Base de Informações de Metadados. Os participantes da rede P2P (*peers*) usam seus navegadores Web como interface para os serviços. A aplicação P2P existente em cada nó é integrada ao navegador Web como um plug-in.

A arquitetura de Park e Wang define também um servidor de políticas (*policy server*). Esse servidor armazena as regras de segurança definidas para a organização e para as comunidades Peer-to-Peer existentes dentro dela. As políticas são armazenadas no formato XML e o RBAC é o modelo escolhido para gerir o controle de acesso.

Toda comunicação entre os participantes da rede P2P e o middleware é realizada através do Gateway de Comunicação, o qual suporta protocolos como o HTTP e o SOAP. Esse componente permite que o Identificador de Serviços receba as requisições dos elementos da rede. O Identificador de Serviços contém documentos WSDL, os quais oferecem uma descrição dos serviços disponibilizados pelo middleware. Na arquitetura elaborada o conceito de Web Services (conjunto de protocolos que visa padronizar a troca de informações entre aplicações) é amplamente utilizado. Os recursos oferecidos pelos participantes da rede são visualizados como “serviços”.

Se um elemento da rede deseja disponibilizar seus recursos para os demais, ele

deve primeiro acessar o middleware e gerar os metadados para esses recursos. No middleware os metadados são criados no Gerador de XML e armazenados na Base de Informações de Metadados. O Gerador de XML também serve para gerar metadados para as políticas de autorização, as quais são depositadas no servidor de políticas.

A base de metadados é distribuída. Um elemento do ambiente P2P utiliza essas informações para localizar o conteúdo que procura. Depois de receber as respostas e analisá-las, o participante escolhe um dos itens da lista para acessar. Nesse momento, o nó requisitor (nó A) procura o middleware e informa a ele que deseja acessar a informação X procurada anteriormente. O middleware então realiza duas funções. Primeiro ele recolhe as informações referentes ao nó A no servidor de políticas e lhe envia esses dados. Segundo, ele pesquisa qual a política de autorização que a organização e a comunidade Peer-to-Peer possuem sobre o recurso X, transforma esta política em metadados XML e envia essas informações ao nó provedor do recurso. Logo após acontece a conexão entre os dois nós da rede Peer-to-Peer. O nó A apresenta suas características e seu papel ao nó B. Este, baseado nessas informações e naquelas enviadas pelo middleware, decide se permite ou não o acesso.

Na arquitetura os papéis utilizados pelo modelo RBAC são as funções exercidas pelos participantes da rede. O sistema toma decisões de controle de acesso baseado na função de cada nó dentro do ambiente Peer-to-Peer e não nas suas identidades. Para suportar o RBAC entre comunidades P2P diferentes, foi elaborada uma ontologia para os papéis do modelo. Por exemplo, o papel Autores na comunidade A e o Editores na comunidade B se referem a mesma função. Os nós podem pertencer a mais de uma comunidade e ter papéis diferentes em cada ambiente.

A arquitetura estipula três tipos de políticas: a política da empresa ou organização (*enterprise*), da comunidade e dos próprios participantes. Cada comunidade pode definir seus próprios usuários, papéis, permissões e associações entre eles. A política de autorização da empresa especifica regras que devem ser seguidas obrigatoriamente por todas as comunidades. Cada nó pode estabelecer a sua própria política, porém ela não deve entrar em conflito com os demais tipos de políticas. Observa-se que existe uma hierarquia de políticas de controle de acesso na arquitetura definida. As políticas da empresa

e da comunidade são centralizadas, enquanto a política que cada nó define é descentralizada, já que cada um deles armazena a sua própria.

O trabalho [PAR 03] evolui com o estado da arte na área de controle de acesso para redes Peer-to-Peer. Os principais aspectos positivos desse trabalho são a utilização do protocolo XML na comunicação entre elementos da arquitetura e a aplicação do modelo RBAC para prover o controle de acesso na rede P2P. Entretanto, ele possui alguns pontos negativos que devem ser destacados. Estes são exibidos a seguir:

- O controle de acesso desenvolvido é realizado sob a arquitetura projetada em conjunto (middleware e seus componentes), ou seja, é específico para esta estrutura. Um projeto de controle de acesso que pretende ser usado por várias aplicações P2P deve escolher um framework Peer-to-Peer sólido e aceito para se basear, como o JXTA que possui código-fonte aberto e apoio da Sun Microsystems. Outros projetos que se fundamentam no JXTA são o TomScop [KAW 04], o qual define um sistema que suporta o trabalho colaborativo em redes P2P, e [HAL 02], que desenvolve um fórum P2P baseado na tecnologia JXTA;

- A arquitetura mescla características de redes P2P puras e híbridas. O middleware, da maneira como está retratado, pode atuar como elemento centralizador no modelo. Um sistema de controle de acesso para redes Peer-to-Peer deve evitar pontos centralizados, pois se este elemento ficar indisponível o sistema perde total ou parcialmente sua funcionalidade;

- A existência de um middleware na arquitetura pode prejudicar a escalabilidade do sistema em geral. Como este middleware centraliza algumas funções críticas, o aumento significativo do número de participantes na rede pode sobrecarregá-lo e comprometer o sistema de controle de acesso;

- Os nós podem estabelecer suas próprias políticas de segurança, porém estes não devem entrar em conflito com as políticas estabelecidas pela comunidade e pela organização a qual pertencem. Os nós precisam confiar na entidade que estipula as políticas para sua comunidade e organização, já que estas regras o atingem di-

retamente (fato não explicitado no artigo descrito). Outro aspecto é que os nós podem perder autonomia, pois não são os únicos a estabelecer permissões para os os recursos que possuem;

- O ambiente em que a arquitetura é definida e desenvolvida é voltado ao meio empresarial (*enterprise*). Um exemplo disso é que os papéis atribuídos aos nós são baseados em suas funções. Modelos de controle de acesso mais genéricos que adaptam-se a diversos ambientes e aplicações P2P podem ser melhor recebidos e implantados em sistemas reais.

5.3 Balanço

A segurança de sistemas Peer-to-Peer é vital para a expansão deste tipo de computação para áreas além do compartilhamento de recursos. As dificuldades para se obter ambientes P2P seguros são várias. Entre as principais estão a disposição geográfica dos participantes, a entrada e saída constante de nós na rede P2P e o caráter descentralizado destas aplicações. Esses fatores dificultam a portabilidade de soluções de segurança testados e aprovados em modelos cliente-servidor para o gênero Peer-to-Peer.

Na pesquisa de segurança em redes P2P alguns temas se destacam: busca da anonimidade nas comunicações existentes em uma rede P2P; construção de relacionamentos confiáveis entre entidades que não se conhecem; controle do comportamento e redução de nós caronas da rede; aplicação de técnicas de reputação e micropagamentos para otimizar a segurança e o funcionamento do ambiente colaborativo; emprego de infraestrutura de chaves públicas para solucionar problemas de identificação e autenticação de nós P2P, entre outros. Este capítulo exibiu o estado da arte das pesquisas nestas áreas.

O capítulo 5 apresentou também os principais trabalhos científicos relacionados ao tema controle de acesso em redes Peer-to-Peer. A arquitetura P2P-Role descrita no próximo capítulo segue as propriedades citadas por [FEN 02], utiliza os resultados obtidos em [PAR 03, BER 04] como experiência e procura avançar o estado da arte na sua área de pesquisa.

Capítulo 6

Arquitetura P2P-Role

O compartilhamento de arquivos e outros recursos entre os pares de entidades abre um enorme potencial para hackers, vândalos e ladrões subverterem os equipamentos envolvidos. A distribuição de músicas na Internet pode não precisar de requisitos de segurança, porém em ambientes P2P corporativos a proteção do canal de comunicação e dos próprios elementos é indispensável. Em particular, a autorização possibilita que cada entidade administre as permissões dos seus recursos e assim coloque em prática a política de controle de acesso que lhe convier.

A autenticação é um requisito prévio para a autorização, ou seja, não existe como definir quais ações ou direitos que uma entidade possui, sem antes garantir a sua identificação. Esta dissertação de mestrado dá ênfase ao processo de autorização; o foco não é desenvolver novos mecanismos de autenticação, mas utilizar aqueles já consagrados na literatura dessa área (por exemplo, o emprego de senhas e certificados digitais).

Este capítulo apresenta a arquitetura de controle de acesso P2P-Role, a qual compõe o núcleo deste trabalho. A ordem dos assuntos é a seguinte: A seção 6.1 descreve o P2P-Role, seu funcionamento e cita os benefícios que ele agrega para as redes Peer-to-Peer. A seção 6.2 discute como as técnicas que incentivam a colaboração podem ser adicionadas à arquitetura definida, a fim de construir um modelo de comportamento para os membros da rede P2P e minimizar a quantidade de caronas que a compõe. Logo após, têm-se a seção 6.3, onde observa-se um cenário de uso onde o P2P-Role e seus

complementos podem ser aplicados.

6.1 Descrição e Funcionamento do P2P-Role

Em geral, as aplicações Peer-to-Peer não empregam nenhum tipo de controle de acesso aos recursos compartilhados na rede. Nesses sistemas, os recursos disponibilizados por um nó podem ser acessados por qualquer outro participante que os encontre, ou seja, não há restrições ou mecanismos que impeçam o acesso às informações. A liberdade e o livre acesso são as características predominantes nessas redes P2P. Contudo, em muitos cenários a existência de mecanismos que reforcem a segurança e confiabilidade é vital. Esse é o caso de aplicações P2P voltadas ao comércio eletrônico entre comunidades, leilões virtuais distribuídos e aquelas usadas entre organizações.

Este trabalho explora esse fato e apresenta a arquitetura P2P-Role, a qual possui foco no controle de acesso e no processo de autorização em redes Peer-to-Peer. O principal objetivo do P2P-Role é o seguinte: desenvolver meios para permitir que os usuários da rede P2P controlem o acesso sob seus recursos e ponham em prática suas políticas de autorização. Conseqüentemente, obtém-se uma rede mais segura e aplicações mais preparadas para agirem em ambientes críticos – aqueles que exigem proteção e robustez.

A Figura 6.1 exibe os elementos que compõem a arquitetura P2P-Role. Os principais componentes são os nós da rede. Cada participante do sistema colaborativo é representado por uma aplicação P2P (igual em todos os nós) e pelo módulo de autorização do RBAC. Os elementos ligados à rede Peer-to-Peer por linhas pontilhadas são opcionais, isto é, seu aparecimento na arquitetura P2P-Role está condicionado ao tipo de rede onde ela é implantada. Eles são o servidor de registro e a autoridade certificadora.

A Figura 6.2 relaciona as particularidades fundamentais do P2P-Role. No P2P-Role cada elemento será o único responsável pela atribuição de permissões aos seus recursos. Sendo assim, a decisão de autorização é individual. Existirão nós da rede utilizando políticas de autorização mais permissivas (por exemplo, permitir acesso irrestrito aos recursos) e outros com políticas mais complexas e restritivas.

O P2P-Role pode ser adequado a redes Peer-to-Peer puras e híbridas. Isso é

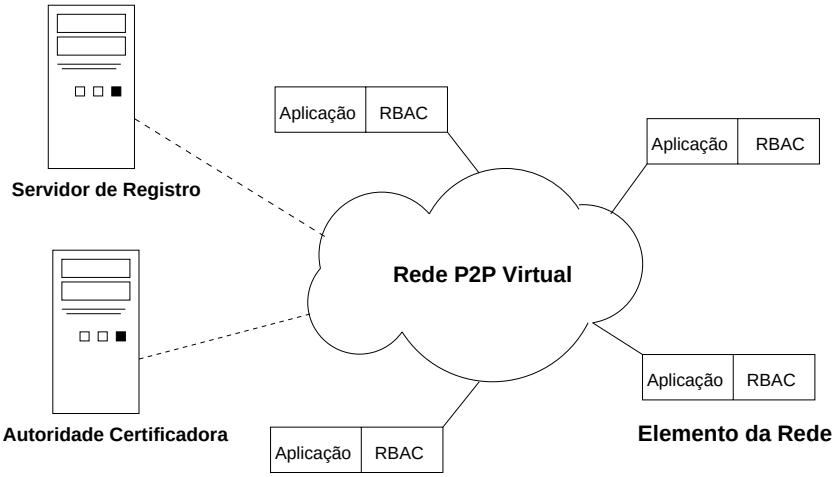

Figura 6.1: Rede colaborativa Peer-to-Peer e o P2P-Role

possível pois a arquitetura definida independe da forma como os nós descobrem quais os elementos da rede que possuem os recursos de que necessitam. O P2P-Role entra em ação depois que um elemento abre uma conexão direta com outro e faz requisições por recursos. O P2P-Role, baseado nas políticas de autorização efetivadas pelo elemento servidor, determina se o acesso à informação desejada deve ser permitido ou bloqueado. Caso o P2P-Role seja utilizado em uma rede P2P pura, o componente servidor de registro não se faz necessário.

A arquitetura P2P-Role optou por empregar o modelo de controle de acesso baseado em papéis – RBAC – como elemento principal nas decisões de autorização, já que ele possui várias vantagens sob os demais modelos desta área (capítulo 4). No P2P-Role cada nó da rede possui seu próprio modelo RBAC. Anexo à aplicação P2P que o nó executa está presente uma interface que possibilita configurar a política de autorização aos seus recursos, ou seja, realizar manutenções nas três entidades básicas do modelo RBAC – usuários, papéis, permissões – conforme necessário.

A anonimidade é uma característica importante das aplicações Peer-to-Peer. Ela permite que exista uma troca de informações entre os usuários da rede P2P sem a identificação explícita dos autores dos recursos, dos nomes dos nós envolvidos na comunicação e do usuário receptor da informação. Conforme [DAS 03], essas peculiaridades são difíceis

de serem atingidas quando os mecanismos de proteção de redes colaborativas P2P são mais rígidos. Esse é o caso do controle de acesso, onde será necessário uma sistemática para identificar (perde anonimidade) claramente nomes de usuários, recursos e nós na rede Peer-to-Peer.

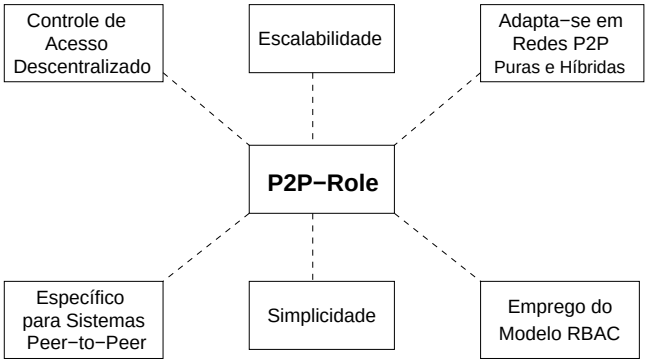

Figura 6.2: Características da Arquitetura P2P-Role

Todos os participantes da rede P2P, em um estado inicial, possuem seus modelos RBAC configurados da mesma forma. O P2P-Role estipula o seguinte padrão para os modelos RBAC presentes nos membros da rede P2P: dois usuários (gerente e anônimo) e dois papéis (especial e comum). A Figura 6.3 exibe a configuração básica do modelo RBAC em cada nó.

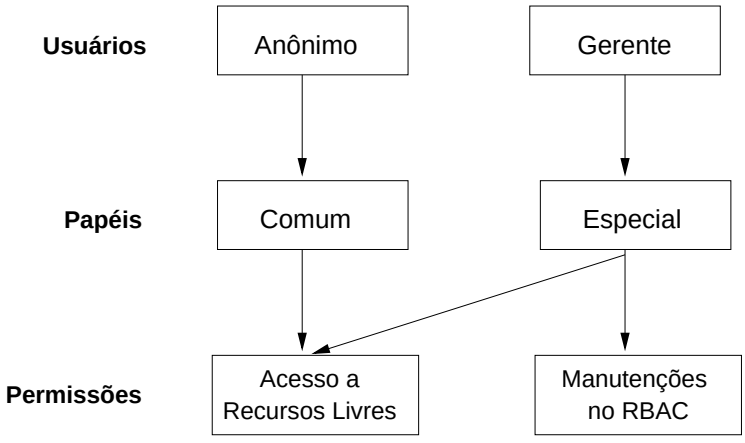

Figura 6.3: Modelo RBAC pré-configurado nos elementos da rede P2P

Os usuários gerente e anônimo estão associados, respectivamente, aos papéis es-

pecial e comum. O papel especial está associado às manutenções do modelo RBAC e possui permissões privilegiadas. Apenas os usuários ligados ao papel gerente estão aptos a modificar a configuração do modelo RBAC do corrente elemento.

O usuário anônimo é associado exclusivamente ao papel comum. Todas as permissões de acesso a recursos considerados livres (qualquer nó pode acessar independente de sua identificação) estão relacionadas ao papel comum.

As entidades do modelo RBAC apresentadas na Figura 6.3 não podem ser excluídas ou alteradas pelos membros da rede P2P. Desta forma, não é permitido, por exemplo, que o usuário gerente do nó crie uma associação entre o papel comum e a permissão que autoriza as manutenções no modelo de controle de acesso (essa ação geraria uma inconsistência de segurança). O gerente do nó P2P pode (e deve), contudo, aperfeiçoar seu modelo RBAC a fim de retratar sua política de segurança. O capítulo 7 (protótipo e experimentos) exibe um exemplo de interface pelo qual os usuários administram seus modelos RBAC.

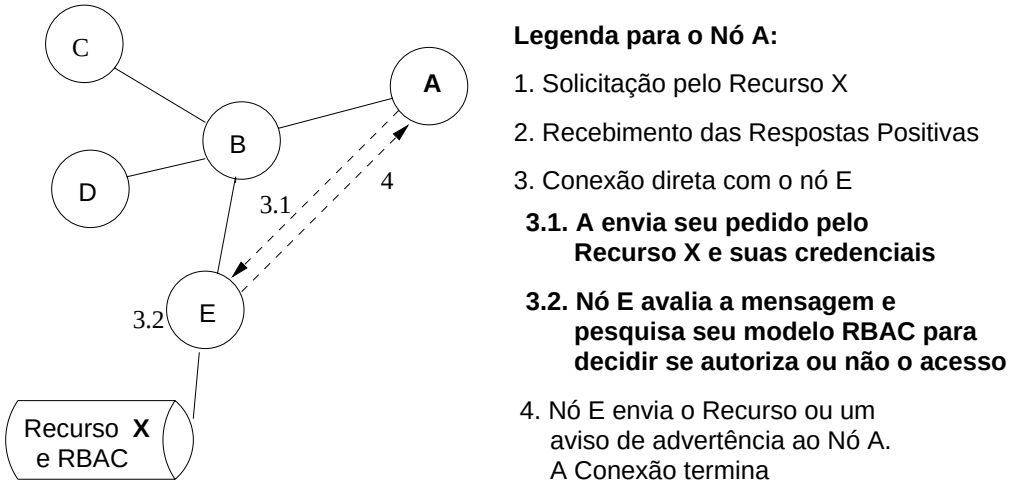

Figura 6.4: Rede Peer-to-Peer pura com P2P-Role

O funcionamento do P2P-Role em uma rede P2P pura acontece em 4 etapas (Figura 6.4): Na primeira etapa o nó lança o pedido por um recurso na rede P2P. Na segunda ele recebe as respostas positivas da rede. Não é escopo do P2P-Role determinar a maneira como o participante procede nos dois primeiros passos – cada tipo de rede P2P aplica seus

próprios métodos. A terceira etapa é realçada na Figura 6.4, pois é nela que o P2P-Role entra em ação. Nesta etapa o nó cliente abre uma conexão direta com o detentor do recurso almejado (passo 3). O nó que recebeu o pedido extrai da mensagem o identificador do recurso e as credenciais do nó requisitor (passo 3.1) e pesquisa o seu modelo de controle de acesso RBAC a fim de definir se a atitude é de negação ou concessão do acesso ao recurso (passo 3.2). Na última etapa (passo 4) o nó servidor envia uma reposta à outra ponta da conexão contendo o recurso ou uma mensagem de advertência informando o motivo do bloqueio do acesso.

A Figura 6.4 mostrou o modo de atuar do P2P-Role em uma topologia P2P pura – sem nenhum elemento central. A Figura 6.5 ilustra a maneira como o P2P-Role é executado em uma rede P2P híbrida. Nessa topologia, a primeira função da aplicação P2P é conectar-se com o servidor de registro, também chamado de catalogador de conteúdo, e passar a ele as informações de cada recurso compartilhado na rede colaborativa (essa etapa é omitida na Figura 6.5). Não é informado nenhum dado sobre as permissões associadas aos recursos. Independente do nível de proteção que o elemento forneça a cada recurso, todos devem ser publicados junto ao servidor de registro.

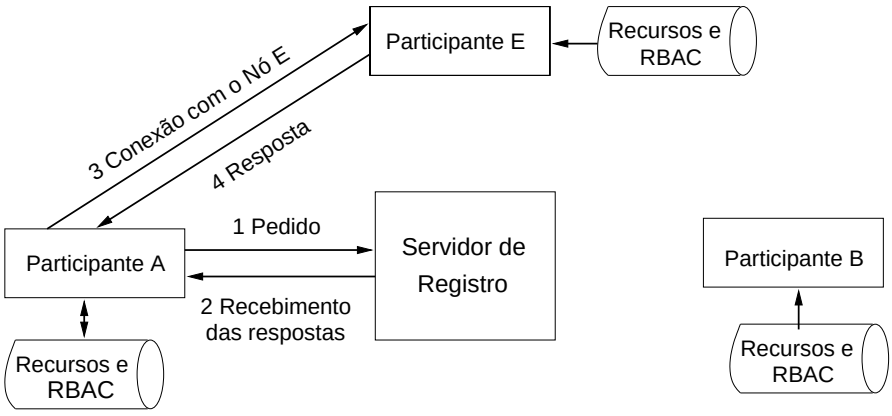

Figura 6.5: Rede Peer-to-Peer híbrida com P2P-Role

Quando um nó da rede deseja uma informação ele questiona o servidor de registro sobre quais outros nós possuem o item procurado. O catalogador de recursos responde com uma lista de nós e o requisitor abre uma conexão direta com algum deles. Na Fi-

gura 6.5 o membro da rede identificado como A criou um canal de comunicação com o participante E; o nó B integra a rede virtual, mas não possui o recurso procurado por A – motivo pelo qual nenhuma conexão foi aberta com ele. A partir do passo 3 nota-se a mesma seqüência de etapas existente na rede P2P pura. O elemento não sabe se terá acesso completo à informação localizada no participante com quem abriu uma conexão, pois a política de autorização adotada pelo nó que possui o recurso é ainda desconhecida para ele.

Quando um nó recebe um pedido por um recurso que detém, ele opta por permitir ou bloquear o acesso a ele. As credenciais do usuário requisitor e o identificador do recurso que ele deseja são passados ao monitor de referência, o qual pesquisa o modelo RBAC implementado no nó. O modelo RBAC verifica suas informações de autorização e decide a situação do acesso. Ele devolve um resultado ao monitor de referência que, por sua vez, comunica a aplicação P2P se a atitude será de concessão ou e negação do acesso. O termo "acessar" pode significar transferência de arquivo para o computador pessoal, observação de conteúdos na tela, etc.

Um usuário de uma rede P2P que emprega o P2P-Role pode demorar algum tempo até encontrar o recurso que deseja e para o qual também tenha permissão de acesso. Logo após lançar a solicitação na rede P2P, diversas respostas lhe são retornadas. Essas respostas informam que determinados nós do sistema possuem o recurso desejado, mas não trazem nenhuma informação de autorização. Sendo assim, o usuário pode ter a experiência de abrir várias conexões até que em alguma delas o direito de acesso lhe seja concedido (autorização aprovada pelo RBAC do nó servidor). Para minimizar o tempo desde um lançamento de requisição até o acesso real à informação procurada, duas otimizações podem ser acopladas ao P2P-Role.

A primeira otimização é a seguinte: no momento de fazer a requisição por um recurso, o usuário pode marcar uma opção chamada "busque apenas recursos com acesso livre". Em uma rede P2P pura, quando um um nó receber essa mensagem ele verifica se possui o recurso identificado e também se seu acesso é livre[1]. O nó apenas emite uma

[1]Todos os recursos considerados livres estão associados ao papel comum. O papel comum esta ligado ao usuário anônimo.

resposta positiva se estas duas condições forem atendidas. O elemento que abriu o pedido sabe que dentre as respostas que obteve, para qualquer uma que escolher, ele terá êxito no acesso.

A segunda otimização também está relacionada com a solicitação de recursos. Normalmente, a mensagem de pedido na rede P2P não possui as credenciais do usuário; o principal item que ela carrega é o identificador do recurso procurado. A otimização consiste em adicionar à mensagem de procura a credencial do usuário que requisita por recursos. Assim, quando determinado participante recebe este pedido, ele verifica se possui o recurso e se a credencial inserida na mensagem possui permissão para acessá-lo. O usuário que iniciou o chamado pelo recurso irá receber respostas dos nós que atenderam às duas exigências descritas.

Um elemento que se agrega à rede virtual P2P, em um primeiro instante, não é conhecido pelos demais membros do sistema. Dessa forma, ele não estará cadastrado em nenhum dos modelos de autorização administrados individualmente pelos participantes da rede. Então, ele terá acesso apenas às informações (recursos) que os demais nós tornam públicas (acesso livre) através da associação da permissão de acesso com o papel "comum" do modelo RBAC. Muitas vezes esse elemento recém chegado à rede irá buscar recursos, encontrá-los e não terá permissão para acessá-los. O P2P-Role determina uma maneira desse usuário tornar-se conhecido pelos demais e, conseqüentemente, fazer parte parte dos modelos RBAC adotados por eles.

No P2P-Role, o usuário de uma aplicação P2P tem a possibilidade de interagir (enviar uma mensagem) com outro elemento da rede, a fim de ser associado/incluso em um papel que possua permissões mais privilegiadas. Nessa parte da aplicação P2P são requeridos 4 itens: nome completo, contato (e-mail), motivo por que almeja o acesso ao recurso e, caso possuir, o certificado digital expedido por uma autoridade certificadora de consenso mútuo dos integrantes da rede virtual. Um exemplo de mensagem pode ser a seguinte: "Nome: Fulano da Silva, Contato: fulano@lrg.ufsc.br, Motivo: Tentei acessar o recurso *segurança.pdf* e a permissão foi negada. Você poderia permitir o acesso a esse recurso para mim. Meu certificado digital, expedido pela CA UFSC, está em anexo".

A forma como um usuário desenvolve o seu modelo de autorização RBAC é des-

conhecida pelos demais participantes da rede. Quando algum integrante faz um pedido a outro nó para ter acesso a uma informação que lhe foi negada anteriormente, ele não precisa conhecer quais papéis e associações que esse nó colocou em prática. Para ele, o que importa é que o nó concorde com a sua reivindicação e aumente seus privilégios.

Outra maneira do usuário comprovar a sua identidade no momento que abre uma conexão com um nó P2P servidor é o uso do certificado digital. Nesse caso não são necessárias as senhas. Basta que cada elemento, no momento do cadastro de novos usuários no seu modelo RBAC, determine que esse usuário será autenticado com base no certificado que apresenta e não com a senha. Observa-se que o certificado digital é importante em dois aspectos no P2P-Role. No primeiro, ele é útil para confirmar que o usuário que solicita inclusão ou alteração no modelo RBAC de outro nó, é realmente quem diz ser. No segundo, ele é utilizado para identificar um nó quando este abre uma conexão direta com outro elemento para acessar os seus recursos.

Com relação ao modelo RBAC, como exposto outrora, cada participante da rede tem uma interface de administração de usuários, papéis e permissões, a qual lhe permite organizar sua política de proteção de recursos. Como o modelo de autorização está acoplado à aplicação sob a forma de módulo (espécie de *plug-in*), o usuário pode mudar de mecanismo de controle de acesso para outro que considerar mais conveniente, mantendo uniforme apenas a chamada feita ao modelo de autorização (Figura 6.6).

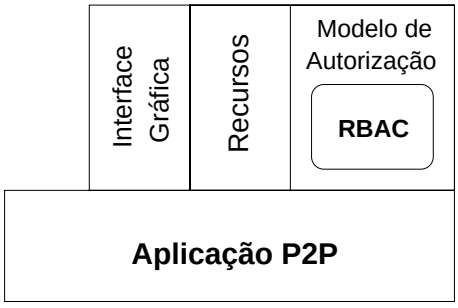

Figura 6.6: Módulo RBAC e a aplicação Peer-to-Peer

6.2 Adição de Técnicas de Incentivo a Colaboração ao P2P-Role

A existência de nós caronas em uma rede P2P degrada o desempenho do ambiente e traz várias conseqüências negativas. O modelo Escambo [RIG 04a] foi desenvolvido para estimular o compartilhamento de recursos na rede e minimizar a quantidade de usuários que apenas sugam o sistema, sem colaborar com nada em troca[2]. O objetivo principal desta seção é apresentar como as funcionalidades do Escambo podem ser adicionadas ao P2P-Role. Pretende-se, com esta união, formar um sistema que regule o comportamento dos nós na rede Peer-to-Peer e possibilite a existência de redes que se sejam, além de seguras, mais justas e harmoniosas.

Esta seção está dividida em duas partes. A primeira parte descreve os aspectos essenciais do modelo Escambo, principalmente seu modo de execução. Logo após, mostra-se os procedimentos necessários para unificar o Escambo e a arquitetura de controle de acesso P2P-Role e a utilidade desta operação.

O modelo Escambo é responsável por controlar o fluxo de recursos entre os participantes da rede Peer-to-Peer. Ele baseia-se na idéia de troca de recursos, onde um nó apenas adquire acesso aos recursos que deseja caso possua outros recursos para disponibilizar no ambiente colaborativo. Dessa forma, participantes que são parasitas - aqueles que não colaboram com os demais - têm acesso restrito às vantagens que a rede proporciona e, então, são encorajados a saírem da condição de caronas para desfrutarem integralmente da comunidade Peer-to-Peer. O modelo Escambo modifica a estrutura normal das redes Peer-to-Peer para alcançar os seus objetivos. Nele, um nó que recebe uma solicitação por um recurso que detém deve, antes de permitir o acesso, verificar se o nó requisitor também dispõe recursos na rede P2P. O Escambo oferece uma sistemática que possibilita reconhecer quais são os nós caronas e quais não são.

Em geral, quando um nó de uma rede P2P recebe uma mensagem de procura por recursos, ele primeiro verifica se possui o recurso e, caso possuir, envia uma resposta posi-

[2]O modelo Escambo foi elaborado de forma paralela ao P2P-Role durante o período do mestrado acadêmico.

tiva. O Escambo define que junto com a resposta, o nó "servidor" deve também comunicar a sua *política de controle de nós caronas*. Esta política informa quais as condições que o nó cliente deve suprir para acessar os recursos deste nó servidor. Ela se divide em três categorias: *(i)* tamanho total dos recursos compartilhados (medido em bytes); *(ii)* número de arquivos oferecidos; *(iii)* tipo de recursos disponibilizados. Um exemplo de política de controle é a seguinte: "os usuários que desejam acesso aos meus recursos devem compartilhar na rede no mínimo 1 Megabyte e 4 arquivos". A Figura 6.7 apresenta os itens que compõem a política de controle de nós caronas.

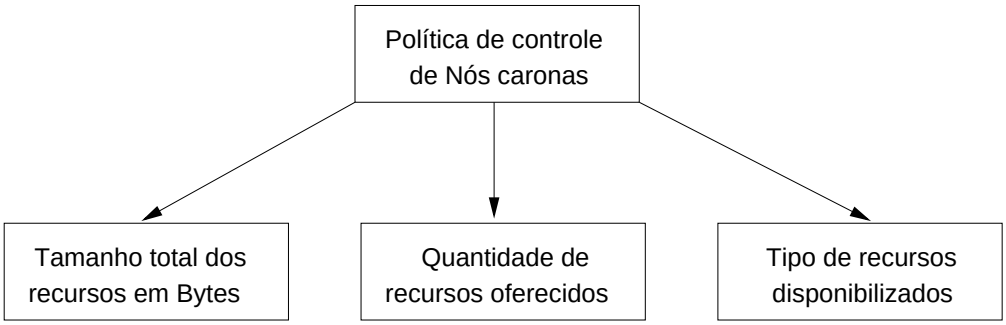

Figura 6.7: Categorias da política de controle de nós caronas

O participante cliente recebe todas as respostas e analisa qual lhe parece ser a melhor. As respostas à procura efetuada carregam junto as políticas de controle de nós caronas adotadas pelos nós que possuem o recurso almejado. Depois de avaliar as políticas que se encaixam em suas características (quais ele pode cumprir), ele abre uma conexão direta com o outro ponto – computação Peer-to-Peer – passando o identificador do recurso que deseja e as características dos recursos que oferece para o ambiente colaborativo. O nó P2P servidor, ao receber a requisição por recurso, verifica se as informações apresentadas pelo cliente P2P satisfazem a sua política de controle de nós caronas e, logo depois, permite ou não o acesso ao recurso.

O Escambo utiliza um modo simplificado de micropagamento [GM 03]. O micropagamento introduz o conceito de pagamento pelo acesso a um recurso ou pedido de atividade. No caso específico do Escambo, o nó cliente paga o servidor através da oferta de recursos na rede Peer-to-Peer.

A reputação nos sistemas colaborativos Peer-to-Peer tradicionais informam quais participantes da rede são honestos ou bom servidores de recursos. A reputação dos nós é adquirida através das experiências dos próprios membros da rede e das trocas de informações de reputação entre os pares que confiam um no outro [MAR 03]. O conceito de reputação encontrado no Escambo distancia-se do mencionado anteriormente. Nele, a reputação de cada nó está associada à quantidade e qualidade do material que ele disponibiliza na rede Peer-to-Peer e não depende da opinião de outros participantes do sistema.

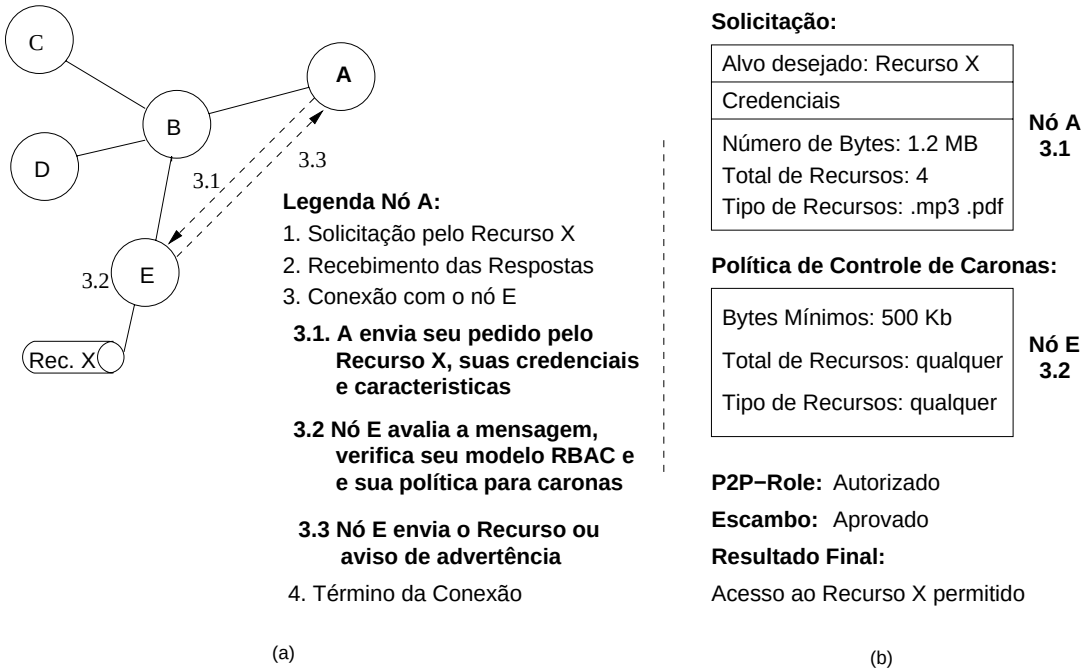

Figura 6.8: Funcionamento do P2P-Role e Escambo integrados

A Figura 6.8(a) apresenta como o modelo Escambo e a arquitetura P2P-Role funcionam integrados. Os dois primeiros passos são responsabilidades da aplicação P2P que adota o P2P-Role e as técnicas de incentivo à colaboração. No passo 3, o nó A seleciona qual nó da rede melhor atende a suas necessidades e faz uma requisição a ele. Como mostra o exemplo, a requisição contém qual o recurso desejado, suas credenciais e as características da sua colaboração para a rede. No passo 3.2 o nó E realiza duas verificações: observa no seu modelo RBAC particular se o nó A possui autorização para

acessar o recurso X e avalia (compara) as características do nó A e sua política de controle de nós caronas. Nesse exemplo o nó A foi autorizado pelo modelo RBAC do nó servidor para acessar o recurso e também foi aprovado no processo de comparação da política de controle de nós caronas e as características dos recursos compartilhados pelo participante cliente (nó A não foi considerado um nó carona).

A Figura 6.8(b) exibe um exemplo de solicitação por recurso e a política de controle de nós caronas aplicada pelo nó E. A requisição elaborada pelo nó A possui o identificador do recurso, suas credenciais – usadas pelo P2P-Role e modelo RBAC – e as características do material que disponibiliza na rede, utilizadas pelo modelo Escambo.

No exemplo anterior o Escambo e o P2P-Role estavam ativos simultaneamente, isto é, para um nó acessar o recurso que deseja ele precisa estar autorizado no modelo de controle de acesso do nó servidor e também comprovar que colabora com recursos para a rede P2P. No entanto, os nós da rede P2P que implementam essa união podem optar por ativar cada um deles separadamente. Por exemplo, um nó pode associar a permissão de todos os seus recursos com o papel comum e usuário anônimo[3] e apenas controlar se os nós que requerem seus recursos são caronas ou não, usando para isso a sua política de controle de nós caronas. A Tabela 6.1 apresenta as opções de execução de uma aplicação Peer-to-Peer que emprega o P2P-Role e o Escambo integrados.

A segurança do modelo Escambo contra a modificação imprópria do protocolo de comunicação por participantes mal intencionados é fundamental. O Escambo estabelece um nível máximo de exigências a ser aplicado na política de controle de nós caronas para evitar, por exemplo, que um nó exija que os membros da rede compartilhem 1GB de dados para acessarem seus recursos. O Escambo também define meios para impedir que usuários alterem o conteúdo das mensagens transmitidas em benefício próprio, fazendo parecer colaborador, quando na verdade não é (detalhe em [RIG 04a]). Tanto o P2P-Role como o Escambo possuem em seus núcleos os conceitos de controle de acesso e autorização, fator que os aproxima. Como mencionado, a união destas pesquisas possibilita o desenvolvimento de sistemas Peer-to-Peer mais protegidos e cooperativos.

[3]O efeito desta atitude é permitir acesso aos recursos independente de suas credenciais.

Opções	Descrição
P2P-Role e Escambo ativos	A aplicação P2P executa com o máximo de proteção. Ela possui ativas as funcionalidades de controle de acesso e incentivo à colaboração.
Somente P2P-Role ativo	Nesta configuração a aplicação executa sem nenhum método para minimizar o número de nós caronas. Apenas o processo de autorização (RBAC) está acionado em cada nó.
Somente Escambo ativo	A arquitetura P2P-Role permanece sem função nessa opção. As técnicas aplicadas pelo Escambo para otimizar a rede são as únicas ativas.
P2P-Role e Escambo inativos	O P2P-Role está inativo quando nenhum controle de acesso é aplicado (todos os recursos estão configurados como livres pelos nós). No Escambo todos os membros adotam a mesma política: permitir conexões independente das características dos recursos disponibilizados pelos clientes P2P.

Tabela 6.1: Opções de funcionamento da aplicação P2P que integra o P2P-Role e o Escambo

6.3 Cenário de Uso

O cenário escolhido para se implantar o modelo resultante da união do P2P-Role e Escambo é uma rede Peer-to-Peer que objetiva a distribuição de códigos-fonte e algoritmos. A qualidade da rede é influenciada pela quantidade de códigos-fonte e algoritmos compartilhados. Quanto maior o número de recursos presente no ambiente, mais usuários são atraídos para a rede e maior é a chance de se encontrar o código-fonte ou algoritmo procurado. A pesquisa desenvolvida auxilia para que haja vários recursos na rede.

A quantidade de códigos-fonte no sistema é inversamente proporcional à porção de nós caronas da rede [PAP 04]. Então, combater os nós caronas implica auxiliar para

uma rede mais justa. O componente Escambo presente em cada nó é utilizado para estimular a cooperação entre os usuários, pois somente aqueles que assistem a rede com códigos-fonte e algoritmos serão aprovados pelas políticas de controle de nós caronas adotadas pelos servidores P2P.

Além do combate aos nós caronas, a arquitetura P2P-Role estendida se preocupa em oferecer aos usuários a possibilidade de classificar os seus códigos-fonte e algoritmos e estabelecer uma política de controle de acesso para eles. Supondo que determinado nó A possui dez códigos-fonte para distribuir, sendo o acesso aos três últimos restrito apenas a uma parcela pequena dos nós da rede. Nesse caso, o nó A pode configurar seu modelo RBAC da seguinte forma:

- associa a permissão de acesso dos sete primeiros códigos-fonte com o papel pré-configurado *comum*. Esta tarefa permite o acesso a esses recursos, independente das credenciais dos clientes;

- cria um papel chamado “Amigos” (exemplo), cadastra os nós da rede Peer-to-Peer que considera amigos e constrói uma associação entre o papel e os usuários (clientes P2P);

- associa a permissão de acesso dos três recursos protegidos com o novo papel criado.

Os nós da rede que não estão vinculados ao papel “Amigo” deste servidor P2P, mesmo se forem caracterizados como colaboradores para rede, terão permissão negada quando tentarem acessar qualquer um dos três últimos recursos compartilhados. O objetivo desta seção foi justificar, através da breve apresentação de uma rede P2P que adota o P2P-Role e seus complementos, os fundamentos principais desta dissertação de mestrado.

6.4 Balanço

O propósito do P2P-Role é aumentar a segurança de redes Peer-to-Peer, proporcionando mecanismos para cada participante do ambiente colaborativo controlar o acesso

aos seus recursos e colocar em execução sua própria política de proteção. Seu foco principal é o processo de autorização. Sendo assim, ele pode ser visto como uma peça de quebra-cabeça que se encaixará com outras para garantir a confiabilidade e a segurança completa de um sistema Peer-to-Peer.

Foram mostrados neste capítulo os componentes que formam o P2P-Role, como acontece a troca de mensagens entre os integrantes da rede e quais os usuários, papéis e permissões configurados originalmente em cada modelo RBAC da topologia. Entre as vantagens mais fortes do P2P-Role estão a sua simplicidade e facilidade de compreensão. Desta forma, ao contrário de outros trabalhos, é simples observar a especificação da arquitetura e reproduzí-la (ou implementá-la) em redes P2P puras ou híbridas e obter os resultados e benefícios mencionados nas seções anteriores. A principal limitação do P2P-Role é que ele pressupõe que a recuperação dos recursos na rede Peer-to-Peer não é fragmentada, ou seja, ele prevê que o download de um recurso seja feito 100% de uma mesma fonte (e não de várias simultaneamente).

Outro assunto descrito no capítulo 6 foi como adicionar as funcionalidades do modelo Escambo ao P2P-Role e torná-lo apto também para combater os usuários caronas da rede Peer-to-Peer. A semelhança preponderante entre o P2P-Role e o Escambo é que os dois estão relacionados a área de controle de acesso (o Escambo indiretamente). Essa característica foi um dos pontos determinantes para a integração entre essas duas pesquisas. Além desse aspecto, outro motivo que influenciou a escolha do Escambo como modelo de incentivo a colaboração para ser acoplado ao P2P-Role foi que nas duas pesquisas a quantidade de trocas de mensagens executadas entre os nós é semelhante, o que facilita a adaptação do P2P-Role para agir também no combate aos usuários caronas da rede Peer-to-Peer.

O tipo de abordagem dada ao P2P-Role neste capítulo foi conceitual. Foram feitas poucas menções sobre a parte prática desta arquitetura. O próximo capítulo apresenta os protótipos construídos durante o desenvolvimento do P2P-Role e permite visualizar como esta arquitetura é inserida em uma aplicação P2P real.

Capítulo 7

Protótipo P2P-Role

Este capítulo apresenta o desenvolvimento do Protótipo P2P-Role, o qual baseia seu método de controle de acesso na arquitetura de autorização definida neste trabalho. O protótipo P2P-Role foi implementado em duas etapas. Na etapa inicial foi elaborada a versão funcional da aplicação P2P-Role. Na segunda fase da implementação vários aspectos do protótipo foram aperfeiçoados, principalmente com a inserção dos projetos JXTA e P2PSockets no seu processo de desenvolvimento.

7.1 Protótipo P2P-Role Original

O objetivo da aplicação P2P-Role[1] é apresentar a viabilidade da arquitetura de controle de acesso construída nesta pesquisa e auxiliar no entendimento de seus conceitos. A discussão sobre esta aplicação é separada em três etapas. Primeiramente, têm-se as principais características que envolvem o protótipo, como o ambiente onde ele foi construído e as decisões de projeto realizadas. Logo após, têm-se o detalhamento do funcionamento do programa P2P e a especificação das mensagens trocadas entre os membros da rede colaborativa. Na seção 7.1.3 encontra-se a definição da interface de administração do sistema RBAC, a qual integra o protótipo P2P-Role.

[1] O protótipo P2P-Role e a arquitetura possuem o mesmo nome. Ele está disponível para no endereço `http://www.lrg.ufsc.br/~rrighi/p2p_role.tar.gz`.

7.1.1 Características do Protótipo P2P-Role

A aplicação Peer-to-Peer elaborada é composta de duas partes distintas. A primeira é responsável por gerenciar os módulos cliente e servidor, os quais compõem o núcleo de cada entidade que participa da rede colaborativa. Já a outra parte possui o objetivo de fornecer ao usuário-administrador de um nó P2P uma forma simples de realizar a manutenção nas tabelas do sistema RBAC e, desta forma, colocar em prática uma política de controle de acesso aos seus recursos.

A linguagem de programação utilizada na escrita do protótipo foi o Java, já que este ambiente de desenvolvimento oferece, através do byte-code (código executável de uma máquina virtual Java), um alto nível de portabilidade de código e suporte a *multithread* nativo (não existe a necessidade de bibliotecas adicionais) [BAR 02]. As tecnologias Java presentes nos módulos elaborados são as seguintes: JDBC, o qual possibilita o acesso uniforme a banco de dados diferentes; *multithread*, para proporcionar a execução de vários fluxos de código simultaneamente e Applet, que viabiliza a composição de programas Java orientados à Web. Conforme [URU 04], a linguagem mais utilizada no desenvolvimento de projetos Peer-to-Peer é o Java.

O protótipo P2P-Role é composto por sete classes: `Application`, `ControlClient`, `ControlServer`, `Auth`, `Message`, `Configuration`, `Admin`. Suas funções estão relacionadas na Tabela 7.1.

As classes `ControlClient` e `ControlServer` estendem a classe `Thread`, ou seja, elas se comportam como fluxos de execução independentes. A classe `Admin` é a única que age como Applet e contribui, assim, para que o responsável pelo nó P2P realize mudanças nos usuários, papéis e permissões aos recursos remotamente.

Em uma aplicação P2P, comumente, o usuário deseja apenas fazer buscas e não disponibiliza nenhum recurso, ou ainda o inverso, onde pretende-se dar exclusividade à oferta de recursos. O protótipo desenvolvido possui a característica de executar o módulo cliente e servidor juntos ou um de cada vez separadamente. Para configurar esse tópico o responsável pelo nó P2P deve editar a classe `Configuration` e alterar a variável apropriada.

Classe	Objetivo
`Application`	É a classe principal da aplicação. É ela quem cria e comanda os fluxos cliente e servidor.
`ControlClient`	Este fluxo de execução realiza a função de cliente e requisita recursos ao fluxo servidor.
`ControlServer`	Recebe os chamados dos clientes, processa-os e devolve a resposta apropriada.
`Auth`	Possui os métodos para avaliar a autenticidade de um usuário e para verificar a autorização aos recursos. Faz chamadas JDBC às tabelas que compõem o modelo RBAC.
`Message`	Especifica as partes que constituem uma mensagem trocada entre as entidades P2P.
`Configuration`	Possui variáveis que definem nomes de arquivos de registro e personalizações da aplicação P2P.
`Admin`	Esta classe é um Applet no qual o responsável pelo nó P2P realiza a manutenção nas tabelas que compõem o modelo de controle de acesso RBAC.

Tabela 7.1: Classes que compõem a aplicação Peer-to-Peer

Uma das particularidades fundamentais do protótipo P2P-Role original é a utilização do nome de usuário e senha (credencial) como mecanismo de autenticação. Ressalta-se, contudo, que a arquitetura de controle de acesso P2P-Role definida no capítulo 6 não está amarrada a um único tipo de mecanismo de autenticação. Outros métodos empregados na comprovação da identidade de um usuário poderiam ser acoplados ao protótipo (por exemplo, certificados digitais) sem que haja alguma modificação no processo de autorização, o qual constitui o núcleo da aplicação P2P-Role.

Outras características existentes são a geração de um registro com todas as conexões bem sucedidas no módulo servidor e a possibilidade de um usuário deixar uma

mensagem para a entidade com a qual ele está se conectando. Como descrito no capítulo 6, nestas mensagens, por exemplo, um usuário pode solicitar sua inclusão em um papel que detenha maiores privilégios, requisição de mudança de senha ou simplesmente enviar um comentário qualquer à outra ponta da conexão . Novamente a classe `Configuration` é a que concentra a especificação dos nomes dos dois arquivos que guardarão as informações citadas anteriormente. Desta forma, os participantes da rede Peer-to-Peer, embora possuam o mesmo código Java em execução, podem ser personalizados diferentemente para refletirem os desejos de seus mantenedores.

O modelo RBAC pré-configurado na aplicação P2P-Role segue aquele descrito no modelo conceitual (Figura 6.3). Os papéis "comum" e "especial" são fixos para todos os elementos. Na arquitetura planejada, as permissões de acesso aos recursos considerados livres (acesso irrestrito) sempre são vinculadas ao papel "comum". Esse é um dos princípios da aplicação P2P-Role.

A Figura 7.1 apresenta a estrutura de diretórios que compõem a aplicação P2P-Role. No diretório `adm` localiza-se a classe `Admin`, utilizada pelo usuário para configurar o seu modelo RBAC. As demais classes encontram-se no diretório raiz da aplicação. No `doc` são armazenados arquivos que orientam o usuário como instalar a aplicação e utilizá-la. O diretório `contabilização` contém os registros das conexões e as mensagens enviadas ao elemento. Por fim, em `recursos` estão todos os arquivos disponibilizados pelo nó para a rede Peer-to-Peer. No P2P-Role, os recursos são arquivos de texto simples.

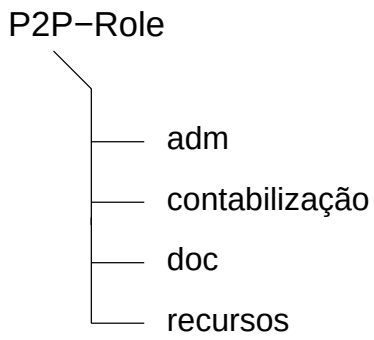

Figura 7.1: Estrutura de diretórios da aplicação P2P-Role

7.1.2 Funcionamento do Protótipo P2P-Role

Como mencionado na Figura 6.4 (capítulo 6), a troca de mensagens na arquitetura P2P-Role acontece em quatro etapas, sendo a terceira a mais importante. As fases são:

1. Solicitação por um recurso na rede;
2. Recebimento das respostas positivas dos demais usuários;
3. Abertura de uma conexão direta com um dos nós que emitiram respostas positivas. O nó servidor avalia a permissão junto ao seu modelo RBAC e verifica se o acesso deve ser liberado ou negado;
4. O nó P2P servidor permite o acesso ou envia uma mensagem de advertência a outra ponta da conexão. Logo após, a comunicação encerra.

O protótipo P2P-Role original não se preocupa com as duas primeiras fases anteriores. Sua finalidade concentra-se em mostrar como acontece o processo de controle de acesso (passo três em diante)[2].

Para colocar a aplicação Peer-to-Peer em funcionamento deve-se executar a classe `Application` e passar como parâmetro o endereço Internet do nó P2P que possui os recursos cobiçados. A aplicação P2P-Role assume que existe uma interface (por exemplo, a Web) onde os nós descobrem quais elementos da rede detém as informações que procuram. A classe `Application` possui duas funções principais. Quando é chamada, esta classe realiza uma varredura no diretório `recursos` e verifica se há algum arquivo novo disponibilizado desde a última vez que a aplicação executou. Se sim, é criado no modelo RBAC do usuário a permissão de acesso a esse objeto. Essa permissão é associada com o papel “comum”. Isso indica que todo recurso oferecido pelo elemento possui originalmente acesso livre. O usuário pode alterar essa situação (restringir o acesso) ou deixar o acesso a esse arquivo aberto aos demais participantes. A outra função da classe `Application` é ativar os fluxos de execução cliente e servidor (por padrão os dois são acionados) e ficar esperando até que eles terminem.

[2]O protótipo P2P-Role Avançado, descrito na seção 7.2, importa-se com as quatro fases da conversação.

O fluxo fornecedor de recursos abre um soquete do tipo servidor e espera por conexões na porta especificada na classe `Configuration`. Já o fluxo cliente conecta-se ao fluxo servidor do nó alvo e requisita o nome de usuário e senha para posteriormente enviá-los para validação. Quando o pedido chega ao fluxo servidor é criado um objeto da classe `Auth` e chamado seu método para a verificação (autenticação) de usuários, repassando a mensagem recém recebida. Este objeto, através de pesquisa nas tabelas que formam o sistema RBAC, retorna se o usuário é válido ou não. Se o processo de autenticação falhar, não é necessário prosseguir com os procedimentos de autorização.

Existem duas possibilidades possíveis de resposta de autenticação: sucesso ou insucesso. Caso acontecer insucesso, o fluxo cliente fecha a conexão e o fluxo servidor retorna a escutar na porta alocada. A opção sucesso gera a gravação de um registro com as particularidades da sessão aberta (*log*) e nesse instante os dois nós que se comunicam estão cientes que a autenticação ocorreu bem e que a etapa inicial da conversa entre eles foi cumprida. Caso o usuário que está abrindo a conexão optar por utilizar o usuário anônimo como credencial, o nó servidor passa pela etapa de autenticação sem fazer verificações e retorna um código de sucesso para o seu par na comunicação. A Figura 7.2(a) apresenta as mensagens trocadas entre os nós que executam o protótipo P2P-Role.

Na próxima etapa da conexão, a aplicação P2P que está realizando a função de cliente faz um pedido por recurso e espera o retorno de seu par na comunicação. O usuário digita o identificador desejado e o cliente envia um objeto `Message` ao servidor contendo o pedido de permissão. Para verificar se o usuário ativo pode acessar determinado recurso, ele invoca o objeto da classe `Auth` e seu método apropriado. Esse objeto usa a seguinte lógica para verificar a permissão: pesquisa todos os papéis associados com o recurso desejado e, logo após, observa se o nome do usuário ativo está ligado a algum desses papéis "com acesso garantido". Se sim, a informação (arquivo com extensão `txt`) é transferida ao usuário que a requisitou.

Na descrição conceitual do P2P-Role não existia uma troca de mensagem específica para a autenticação. O processo de autenticação e autorização aconteciam juntos, ou seja, o usuário após localizar qual nó detém o recurso que lhe interessa abre uma conexão com ele passando de uma vez só as suas credenciais e o recurso desejado. O nó

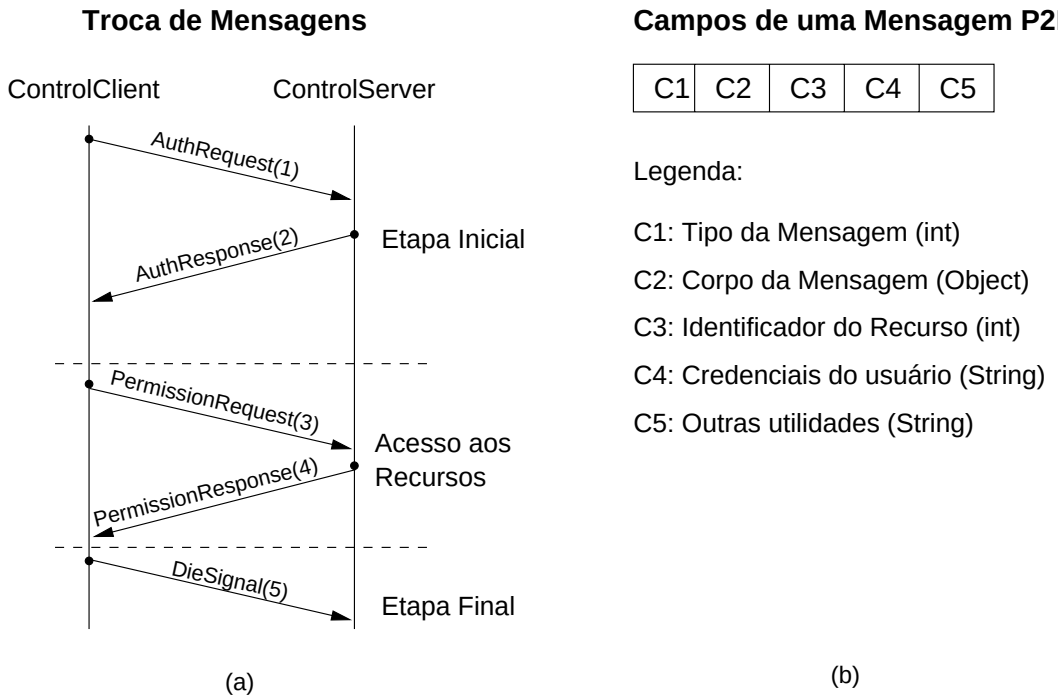

Figura 7.2: Mensagens trocadas entre os membros da rede P2P

servidor verifica a autenticidade das credenciais e se elas possuem acesso liberado ao recurso preenchido na mensagem. Não existe uma resposta para autenticação e outra para a autorização – apenas uma para os dois. Observa-se, portanto, que o protótipo P2P-Role original diferencia-se um pouco da especificação da arquitetura P2P-Role, porém continua a seguir seus princípios.

Da maneira como está implementado o P2P-Role, se o usuário desejar dois recursos de um mesmo nó servidor ele não precisa realizar duas vezes a fase de autenticação (passos 1 e 2 da Figura 7.2(a)). Logo depois de obter o resultado do acesso ao recurso requisitado, o fluxo cliente fica à espera de outro identificador de recurso (emitir outra mensagem `PermissionRequest`) ou que o usuário feche a conexão com este nó servidor e volte para a tela principal da aplicação.

A organização das classes desenvolvidas e o fluxo de comunicação entre os elementos estão presentes na Figura 7.3. É importante observar que uma conexão em busca de recursos parte sempre da classe `ControlClient` (função de cliente) em direção à

classe `ControlServer` (função de servidor). A classe `Auth`, que possui métodos voltados à autenticação e autorização de usuários, somente é utilizada pelo fluxo de execução que serve recursos. Outro aspecto relevante é que a classe `Admin`, presente em cada elemento, não aparece na Figura 7.3. Ela é executada separadamente da aplicação P2P principal e seu objetivo, como mencionado, é possibilitar a administração do modelo RBAC de cada participante da rede. A seção 7.1.3 discute essa questão específica.

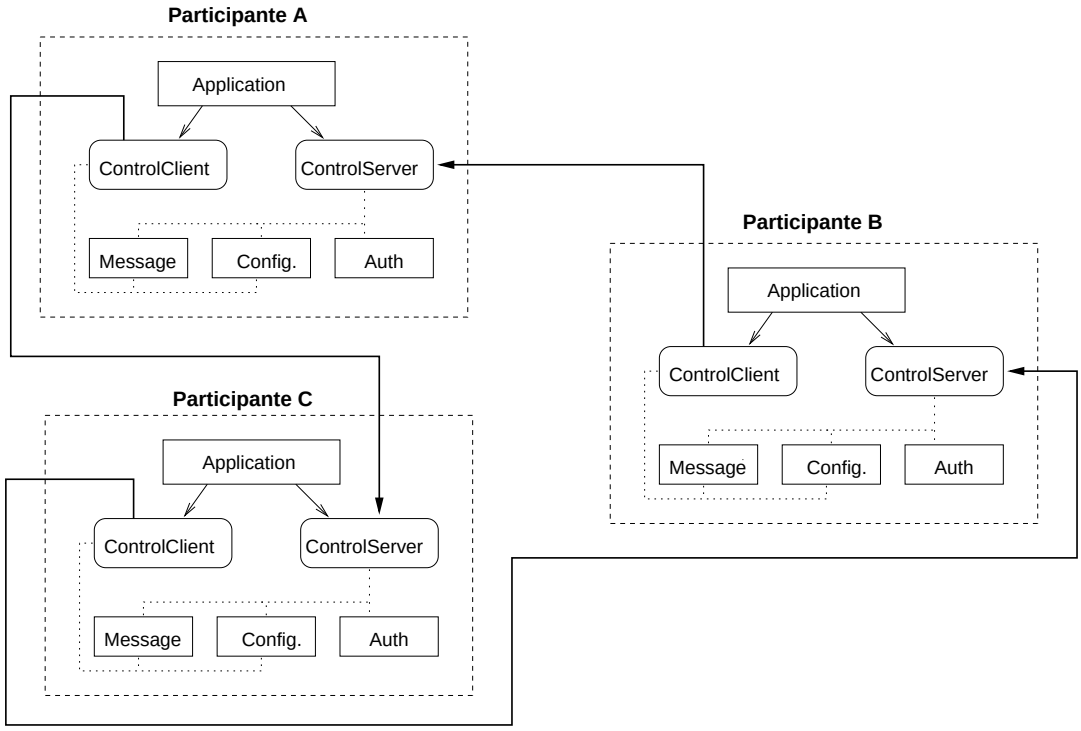

Figura 7.3: Comunicação entre os elementos da rede Peer-to-Peer

As outras possibilidades que o usuário possui, além de requisitar recursos, são as de emitir mensagens simples ao nó servidor com quem está se comunicando (os propósitos deste tipo de mensagem foram exibidos na página 91) e fechar uma conexão ativa. Se o item escolhido é o fechar (identificado pelo número 999), o cliente envia um objeto `Message` informando que vai "morrer"; fato que possibilita que ambas partes fechem juntas a conexão ativa e nenhum problema aconteça com os soquetes. A aplicação P2P, nesse ponto, questiona ao usuário se ele pretende abrir uma conexão diferente com outro nó da rede colaborativa ou terminar o fluxo cliente. Caso outra conexão seja aberta, todo

o ciclo de conversação recomeça.

A aplicação P2P-Role inicial baseia-se no endereço IP para identificar os participantes da rede P2P. Essa característica fere o princípio da anonimidade da comunidade Peer-to-Peer, porém favorece o mantenedor da aplicação que saberá, através dos registros do P2P-Role, qual a procedência dos usuários que buscaram seus recursos. O trabalho desenvolvido por Marti [MAR 03] aborda o tema anonimidade em sistemas colaborativos e cita métodos alternativos ao endereço IP para serem usados na identificação de usuários[3]. Um desses métodos é o uso de certificados digitais em conjunto com uma infraestrutura de chaves públicas.

7.1.3 Administração do Modelo de Controle de Acesso RBAC

O sistema de controle de acesso existente em cada nó da rede Peer-to-Peer é o RBAC. No protótipo construído este sistema foi idealizado em seis tabelas do banco de dados MySQL (versão 4.01), com destaque para as seguintes: usuário, papel, direito e hierarquia_papel. As tabelas restantes são o resultado do relacionamento que há entre os usuários e papéis e entre os papéis e os direitos. O módulo de administração é o responsável por gerenciar o conteúdo existente nessas tabelas e assegurar que os propósitos de segurança planejados por determinado membro da rede sejam obedecidos. A Tabela 7.2 relaciona as seis tabelas criadas no banco de dados e suas descrições.

Para realizar tarefas administrativas é necessário uma autenticação prévia. Através dessa etapa são verificados dois requisitos: se o nome do usuário e a senha conferem e se este usuário está associado ao papel “Especial”. Caso essas exigências sejam satisfeitas, o usuário-administrador tem acesso à barra de menus do programa, a partir do qual ele fará suas tarefas. A Figura 7.4 exibe a tela apresentada ao administrador do nó nesse instante.

A barra de menus é composta de sete itens: usuário, papel, direito, usuário-papel, papel-direito, hierarquia e ajuda. Seus nomes informam ao administrador como atingir o

[3] A anonimidade não é o objetivo principal do P2P-Role original, motivo pelo qual esta aplicação utiliza um mecanismo simples (IP) para a identificação dos nós da rede.

Nome da Tabela	Descrição
`usuario`	Contém informações sobre os usuários cadastrados, como seu nome completo, *login* e *hash* da senha.
`papel`	Armazena os papéis do modelo RBAC. Todos os papéis possuem uma descrição e uma data de criação.
`direito`	Guarda as permissões do sistema. Na aplicação P2P-Role, um usuário que deseja cadastrar a permissão de acesso ao recurso de nome `x.txt` deve preencher o campo recurso dessa tabela com o caminho completo do arquivo. Pode-se estabelecer uma descrição para cada permissão.
`usuario_papel`	Essa tabela relaciona os identificadores dos usuários e dos papéis. A tupla X, Y informa que o usuário com identificador X é vinculado ao papel representado por Y.
`papel_direito`	Relaciona os identificadores dos papéis e dos direitos. A atribuição de permissões aos papéis é feita nessa tabela.
`hierarquia_papel`	Essa tabela permite o uso de hierarquia entre os papéis. Ela possui dois campos: papel-pai e papel-filho.

Tabela 7.2: Tabelas que compõem o modelo RBAC de cada elemento

seu propósito, ou seja, se for necessário incluir um novo usuário, o menu “usuário” deve ser procurado.

Os menus Usuário, Papel, Direito possuem os mesmos nomes de opções. Os três possibilitam as tarefas de adicionar, remover e listar os registros das tabelas do banco de dados MySQL que formam o sistema RBAC. Os outros menus são responsáveis por estabelecer as associações entre os registros do modelo de acesso (por exemplo, para associar o usuário “anonimo” com o papel “comum” deve-se dirigir ao menu Usuário-Papel) e por estabelecer as relações de “pai e filho” entre os papéis (por exemplo, papel Funcionário é pai do papel Professor). O último menu chama-se Outros e disponibiliza informações sobre o RBAC, além de conter o item “Fechar”. Ao executar esse procedimento, o ge-

Figura 7.4: Módulo de administração do RBAC

rente encerra sua sessão de trabalho e a tela de autenticação novamente fica a espera de um usuário.

Como relacionado, cada integrante do ambiente P2P possui seu próprio modelo RBAC, isto é, existe um conjunto de tabelas para cada membro da rede. A Figura 7.5 apresenta o diagrama Entidade-Relacionamento (ER) existente entre entidades do modelo RBAC desenvolvido. Nota-se que um usuário pode estar associado com N papéis, assim como um papel pode estar associado a vários usuários. De maneira semelhante, uma permissão pode estar ligada a vários papéis e um papel pode ser vinculado a vários

direitos.

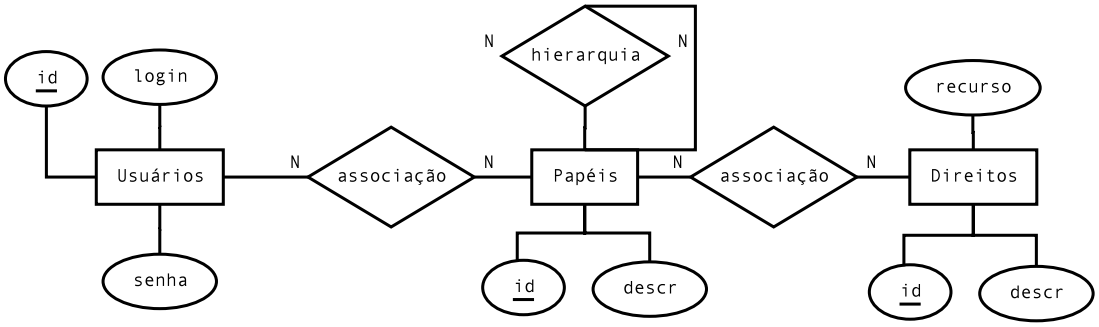

Figura 7.5: Diagrama Entidade-Relacionamento desenvolvido

7.2 Protótipo P2P-Role Avançado

O objetivo desta seção é apresentar os aprimoramentos feitos no protótipo P2P-Role a fim de torná-lo mais completo. O complemento mais importante adicionado foi a utilização do projeto P2PSockets, o qual permite a escrita de sistemas Peer-to-Peer mantendo-se a mesma API[4] dos soquetes TCP/IP existente no Java. A oportunidade de reutilizar códigos escritos anteriormente e de usufruir das vantagens do JXTA transparentemente são as principais justificativas da escolha do projeto P2PSockets.

Além do uso do P2PSockets no desenvolvimento da aplicação P2P-Role, outras melhorias foram executadas. Elas estão listadas a seguir:

- implementação de uma aplicação P2P completa. A procura por recursos e o recebimentos das respostas foram elaborados;
- possibilidade de atendimento paralelo de clientes P2P; vários podem acessar recursos em determinado elemento no mesmo instante;
- desenvolvimento de um módulo que permite a visualização gráfica de modelos RBAC chamado RView.

[4] *Application Programming Interface*

As sete classes Java construídas no protótipo P2P-Role original foram mantidas nesta nova etapa de construção da aplicação. As classes `Application`, `Control-Client`, `ControlServer` e `Configuration` sofreram algumas modificações que objetivam adequá-las ao P2PSockets e às melhorias realizadas. Foram criadas duas novas classes denominadas `InformStatus` e `HandleUser`. A classe `InformStatus` possui duas atribuições: informa continuamente o estado da aplicação a uma entidade que cataloga os recursos presentes na rede e também descobre, através de uma pesquisa junto a esta entidade, quais participantes da rede possuem os recursos desejados pelo usuário que executa a aplicação. A classe `HandleUser` é um fluxo de execução independente (*thread*) que é chamado para tratar cada pedido por recurso separadamente. A Figura 7.6 mostra como um nó da rede está organizado depois que foram aplicadas as modificações ao P2P-Role. As duas novas classes estão inseridas no conjunto representado.

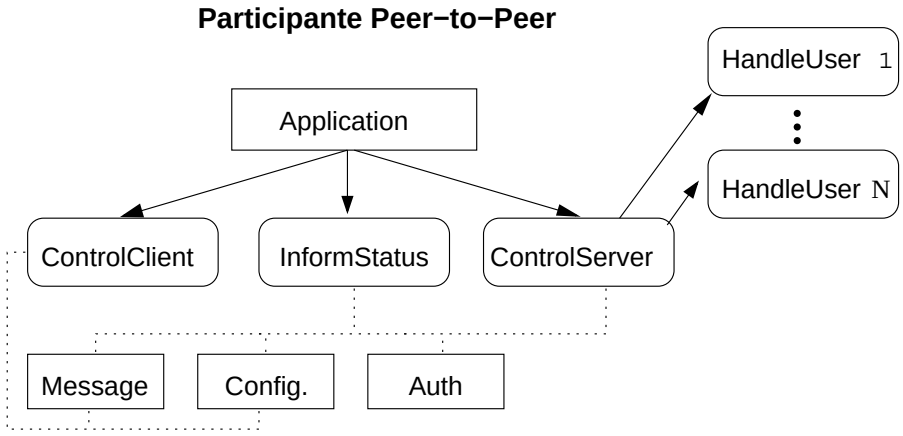

Figura 7.6: Organização das classes no P2P-Role aperfeiçoado

O novo P2P-Role não utiliza o endereço IP para identificar os nós da rede P2P. Ao invés disso ele usa o nome completo do gerente do nó (por exemplo, Rafael da Rosa Righi) para distinguir cada elemento do ambiente. Cada usuário deve preencher um atributo da classe `Configuration` chamado *identification*. Ele será utilizado no momento de abrir um soquete do tipo servidor P2P. O código a seguir compara como a classe `ControlServer` abre um soquete do tipo servidor no P2P-Role original e no atual. Observe que os dois produzem um objeto do mesmo tipo – *ServerSocket*.

```
// Sem o P2PSockets. A porta de escuta é 2004.
java.net.ServerSocket ServerAplication =
                      new ServerSocket(2004);

// Com o P2PSockets. Porta de escuta é 100.
// 'identification' possui o nome completo do gerente nó.
java.net.ServerSocket AdvancedServer  =
                      new P2PServerSocket(
                       Configuration.identification, 100);
```

Com relação ao código anterior, uma das principais questões a serem respondidas é como a classe *ControlClient* consegue encontrar e se conectar a um nó na rede P2P tendo conhecimento apenas de seu nome completo e a porta em que ele espera por conexões. A resposta para essa questão é a seguinte: quando um nó abre um soquete servidor P2P, o P2PSockets internamente informa a um nó *rendezvous* (conceito do JXTA) configurado no código do projeto[5] a identificação do nó, o grupo a que ele pertence – estabelecido na classe `Application` – e a porta que ele utiliza. Assim, quando um nó abre um soquete do tipo cliente P2P, o P2PSockets contata o mesmo nó *rendezvous* e o questiona sobre a existência de um nó representado por determinado "nome completo".

A estrutura do protótipo P2P-Role avançado está apresentada na Figura 7.7. O P2P-Role utiliza os serviços do projeto P2PSockets que, por sua vez, usufrui das características da tecnologia JXTA.

A tecnologia JXTA utiliza o protocolo XML em diversas situações, entre elas a comunicação entre os elementos da rede P2P. O protótipo P2P-Role não foi codificado com suporte a XML, mas desfruta deste recurso indiretamente. Isso acontece devido a utilização do P2PSockets no desenvolvimento do P2P-Role avançado e a sua sustentação no JXTA.

A classe `Auth` possui os métodos para tratar os procedimentos de autorização. Ela foi mantida intacta durante as transformações que ocorreram desde o P2P-Role origi-

[5]Para o desenvolvedor é irrelevante o endereço IP público deste computador.

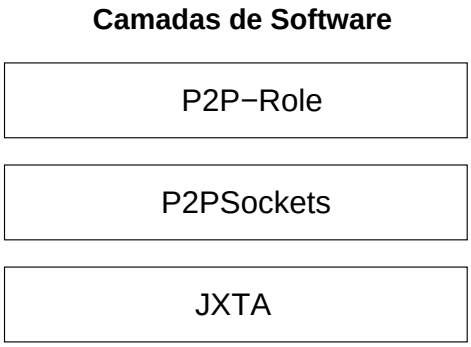

Figura 7.7: Estrutura em camadas do protótipo elaborado

nal. Então, a forma como um nó decide se deve autorizar ou não o acesso sob um recurso descrita na seção 7.1.2 (funcionamento do P2P-Role) também é válida nesse novo contexto. A troca de mensagens realizada depois que um elemento verifica quem possui o recurso que ele procura (Figura 7.2) continua a mesma.

A Figura 7.8 expõe as possibilidades que um usuário possui no protótipo P2P-Role. Entre elas está a visualização gráfica de seu modelo RBAC. Esse tema é discutido na seção seguinte.

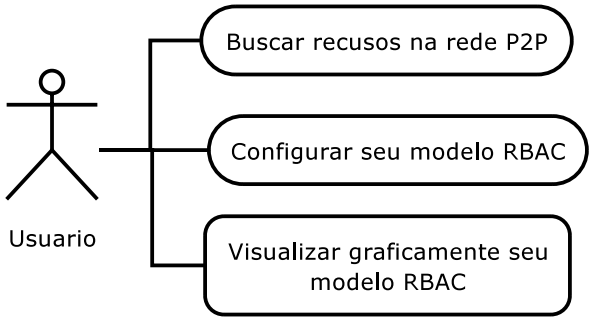

Figura 7.8: Casos de uso da aplicação P2P-Role

7.2.1 Módulo RView - Visualização Gráfica de Modelos RBAC

Os usuários de cada aplicação P2P configuram seus modelos RBAC através de um Applet Java específico para esse fim (Figura 7.4). Nele os integrantes da rede efetivam a

política de autorização que escolheram. O RView[6] permite que os usuários observem de maneira gráfica os modelos RBAC que projetaram.

O RView fundamenta-se em uma biblioteca gráfica chamada JGraph[7]. Ele lê as informações do modelo RBAC de alguma fonte, interpreta-as e gera o gráfico correspondente. O RView pode ler os dados do modelo RBAC de duas formas: *(i)* com varreduras nas tabelas do MySQL criadas em cada nó; *(ii)* através da compreensão de um arquivo XML que descreve a política de controle de acesso. Abaixo está exposto um trecho de um arquivo XML entendido pelo RView.

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE rbac SYSTEM "rbac.dtd">
<rbac>
      <usuario id='u1'>
         <u_nome> Gerente </u_nome>
         <u_senha> 202cb9075b964b07152d234b70 </u_senha>
         <u_connect>
             <u_papel id = 'p1'/>
         </u_connect>
      </usuario>

      <papel id='p1'>
         <p_nome> Especial </p_nome>
         <p_descricao> Responsável Sistema </p_descricao>
         <p_connect>
             <p_direito id='d1'/>
             <p_direito id='d3'/>
         </p_connect>
      </papel>
</rbac>
```

[6] http://pet.inf.ufsc.br/~rview

[7] http://www.jgraph.com

A utilização do RView não se restringe às redes P2P. Desenvolvedores e interessados em segurança computacional podem utilizá-lo em seus sistemas ou para outros fins, como o ensino de controle de acesso em universidades e organizações.

A Figura 7.9 apresenta um modelo RBAC representado pelo RView. O RView apresenta os modelos RBAC da seguinte forma: três linhas principais, sendo uma para cada entidade do RBAC – usuário, papel e permissão; cores diferentes para cada linha, a fim de salientar a divisão deste modelo de autorização em três entidades principais; setas que constroem funções de relacionamento entre os componentes representados. A forma de representação de modelos RBAC adotada pelo RView baseia-se na pesquisa de Koch e Mancini [KOC 02], que estuda como a teoria de grafos pode ser aplicada ao modelo de controle de acesso RBAC para auxiliar na sua visualização.

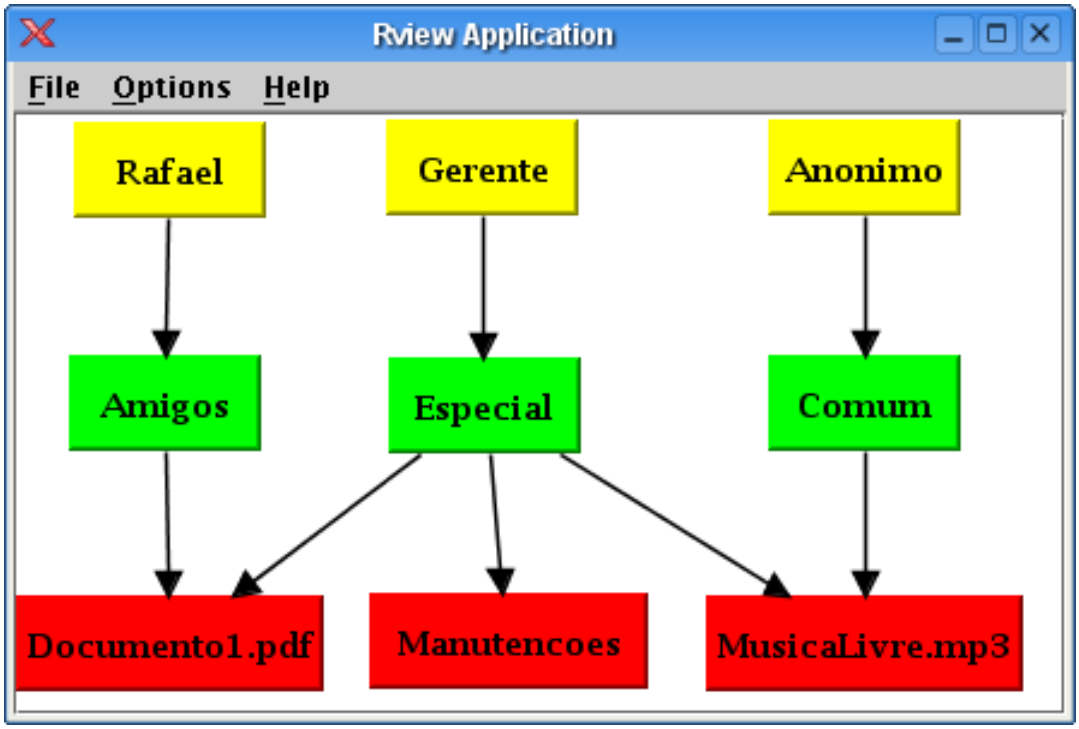

Figura 7.9: Exemplo de modelo RBAC representado no RView

7.3 Balanço

O protótipo P2P-Role surgiu simples e foi sendo aperfeiçoado durante o andamento das pesquisas. Este capítulo foi responsável por retratar suas particularidades, funcionamento e vantagens. A reescrita do P2P-Role original com o P2PSockets agregou qualidade à aplicação e trouxe a oportunidade de nós localizados em redes protegidas (aquelas que adotam *firewalls* e tradutores de endereços de rede) usufruírem integralmente do ambiente Peer-to-Peer.

O modelo RBAC construído para cada elemento da rede P2P é do tipo RBAC1 (capítulo 4), o qual possui suporte à hierarquia de papéis. Com essa característica os gerentes dos nós podem efetivar políticas de autorização mais completas e sofisticadas.

O protótipo P2P-Role não implementa fatores do modelo Escambo. A parte do capítulo 6 que revela como o Escambo e o P2P-Role funcionam integrados não foi o foco do desenvolvimento. O capítulo atual e o anterior compõem o núcleo dessa dissertação. Eles apresentam juntos os aspectos teórico e práticos da arquitetura de controle de acesso P2P-Role. Este trabalho termina no próximo capítulo com a Conclusão e com a descrição dos possíveis complementos para a pesquisa efetuada.

Capítulo 8

Conclusão

As redes Peer-to-Peer possibilitam a colaboração entre os elementos de rede e seu potencial é imenso, desde o compartilhamento de recursos até o comércio eletrônico entre comunidades. As aplicações P2P precisarão ser seguras e confiáveis para atingirem as corporações e ambientes críticos. Nas redes Peer-to-Peer cada elemento disponibiliza seu conjunto de recursos para os demais membros da rede virtual. Normalmente, esses recursos estão acessíveis a todos e não existe distinção entre elementos do sistema P2P. Porém, em algumas situações os nós podem desejar limitar o acesso a determinadas informações, sendo necessária a adoção de um mecanismo de controle de acesso que contemple o paradigma de ambientes colaborativos P2P.

Este trabalho concentrou-se na proteção de redes Peer-to-Peer e no desenvolvimento de uma arquitetura de controle de acesso própria para esse gênero de sistemas. A arquitetura P2P-Role possibilita que cada participante do ambiente colaborativo administre individualmente a política de segurança dos seus recursos. Sendo assim, cada nó será o único responsável por determinar quais as entidades que poderão acessar determinado recurso sob seus cuidados. Dentre as decisões de projeto realizadas no decorrer da elaboração do P2P-Role está a opção pelo modelo de controle de acesso baseado em papéis – RBAC.

O modelo RBAC utiliza o conceito de papéis para intermediar o processo de vinculação de permissões aos usuários em um sistema computacional. Essa característica

auxilia para a flexibilidade deste modelo e facilita o processo de elaboração de regras de controle de acesso. No P2P-Role foi definido que a configuração inicial do modelo RBAC é igual para todos os integrantes da rede Peer-to-Peer. Entretanto, cada nó é livre para acrescentar suas próprias informações de autorização e criar novos usuários, papéis e permissões, conforme necessário.

Esta dissertação também apresentou como as técnicas que incentivam a colaboração podem ser agregadas à arquitetura P2P-Role a fim de regular o comportamento dos usuários e otimizar o funcionamento de uma rede Peer-to-Peer. A abordagem empregada é nova, pois se estabelece uma relação entre dois temas importantes dentro do paradigma Peer-to-Peer e os benefícios de funcionarem integrados.

O modelo de incentivo à colaboração explorado no decorrer desta pesquisa foi o Escambo. Ele adota uma função de reciprocidade, onde declara-se que apenas nós que dispõem recursos no ambiente colaborativo podem desfrutar de suas vantagens integralmente. No Escambo estão presentes conceitos de reputação e micropagamentos. Esse modelo aplica um modo diferenciado de micropagamento, pois os nós clientes “pagam” os elementos servidores através da oferta de recursos na rede P2P.

A união das funcionalidades do P2P-Role e do Escambo cria um modelo de comportamento, o qual aplica regras de conduta aos nós que compõem a rede Peer-to-Peer de forma semelhante ao processo descrito em Strulo [STR 04]. O desenvolvimento do P2P-Role priorizou, diferentemente de Park e Wang [PAR 03], tornar simples a tarefa de configuração do modelo de controle de acesso existente em cada nó da rede P2P. A intenção é que a aplicação P2P agregue segurança, mas mantenha a sua facilidade de utilização.

O protótipo P2P-Role auxilia para o entendimento dos conceitos existentes na arquitetura de controle de acesso estabelecida e na visualização de suas características. Seu modelo de classes e métodos pode servir para nortear a escrita de outras aplicações P2P, principalmente aquelas que exigem segurança. Foram utilizados no processo de implementação os projetos JXTA e P2PSockets. Estes projetos forneceram a base para a construção do protótipo P2P-Role e possibilitaram a manutenção do foco da aplicação no controle de acesso e não na reescrita de uma nova infra-estrutura de comunicação

Peer-to-Peer.

O P2P-Role possibilitou verificar que a implantação de um controle de acesso na comunidade P2P pode prejudicar a anonimidade inerente a esses sistemas distribuídos. Assim, projetistas de aplicações colaborativas podem constatar que o reforço dos mecanismos de segurança nessas redes geralmente vêm acompanhado de restrições nas principais características desses sistemas.

O desenvolvimento da aplicação RView foi outro aspecto importante desta dissertação. Ela possibilita que nós da rede visualizem graficamente seus modelos RBAC; fator que auxilia para não haver manutenções equivocadas nas informações de autorização, já que os possíveis erros seriam facilmente detectados quando o usuário observasse seu modelo RBAC na forma gráfica. Além dessa função, o RView pode ajudar (assessorar) no ensino do modelo de controle de acesso baseado em papéis e em questões relacionadas à proteção e segurança de sistemas.

Como se pode notar, as contribuições desta dissertação foram: desenvolvimento de uma arquitetura de controle de acesso específica para sistemas Peer-to-Peer; adição de técnicas que incentivam a colaboração à arquitetura definida; implementação baseada nos projetos JXTA e P2PSockets; elaboração de um módulo para a visualização gráfica de modelos RBAC. Outra colaboração importante foi a realização de um amplo levantamento bibliográfico a respeito da segurança em redes Peer-to-Peer.

Com relação a trabalhos futuros, indica-se três opções. Elas estão relacionadas a seguir:

- A primeira é a união do protótipo P2P-Role à infraestrutura de chaves públicas e ao modelo de certificados digitais;
- Codificar no protótipo P2P-Role a especificação do modelo Escambo, baseando-se no modelo conceitual (capítulo 6) que menciona como a arquitetura de controle de acesso P2P-Role e as técnicas que estimulam a colaboração são integradas em um sistema Peer-to-Peer. O protótipo P2P-Role atual não se preocupou com as extensões da arquitetura desenvolvida;
- A última opção baseia-se no fato que existem várias pesquisas em segurança de

redes Peer-to-Peer e que cada uma delas trata de uma tema em particular (anonimidade, autenticação, autorização, controle de nós caronas, outros). Portanto, o desenvolvimento de um sistema que unifique os resultados obtidos por esses estudos determina um modelo que concentra vários requisitos de segurança P2P e valoriza o que existe de melhor em cada uma dessas pesquisas.

Referências Bibliográficas

[ADA 00] ADAR, E.; HUBRMAN, B. Free Riding on Gnutella. **First Monday**, v.5, n.10, p.1–14, Outubro, 2000.

[AGR 03] AGRE, P. E. P2P and the Promise of Internet Equality. **Communications of the ACM**, v.46, n.2, p.39–42, 2003.

[AND 01] ANDERSEN, D. et al. Resilient overlay networks. **In: Eighteenth ACM Symposium on Operating Systems Principles**, Banff, Alberta, Canada, p.131–145, Outubro, 2001.

[BAL 03] BALAKRISHNAN, H.; KARGER, D.; MORRIS, R. Looking Up Data in P2P Systems. **Communications of ACM**, v.46, n.2, p.43–48, Fevereiro, 2003.

[BAR 01] BARKAI, D. Technologies for Sharing and Collaborating on the Net. **In: Proceedings of the First International Conference on Peer-to-Peer Computing (P2P'01)**, Lingköping, Sweden, p.13–28, Agosto, 2001.

[BAR 02] BARCELOS, M. Programação paralela e distribuída em java. **In: II Escola Regional de Alto Desempenho (ERAD 2002)**, São Leopoldo, RS, p.179–181, Janeiro, 2002.

[BEL 76] BELL, E.; LAPADULA, L. J. Security Computer Systems: Unified Exposition ad Multics Interpretation. BedFord, MA: MITRE Corporation, Março, 1976. Relatório Técnico.

[BER 04] BERKET, K.; ESSIARI, A.; MURATAS, A. PKI-based Security for Peer-to-Peer Information Sharing. **In: Proceedings of Fourth IEEE International Conference on Peer-to-Peer Computing (P2P'04)**, Zurich, Switzerland, p.45–52, Agosto, 2004.

[BRO 04] BROIDO, A. et al. Transport Layer Identification of P2P Traffic. **In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement**, Taormina, Sicily, Italy, p.121–134, Outubro, 2004.

[BUE 03] BUENO, C.; LOPES, N. Towards Peer-to-Peer Content Indexing. **ACM SIGOPS Operating Systems Review**, v.37, n.4, p.90–96, Novembro, 2003.

[CAV 04] CAVIGLIONE, L. The "Dark Side" and The "Force" Of The Peer-to-Peer Computing Saga. **Peer-to-Peer Journal (P2PJ)**, v.1, n.4, p.1–11, Janeiro, 2004.

[CHA 04] CHAVDA, K. F. Anatomy of a Web Service. **Journal of Computing Sciences in Colleges**, v.19, n.3, p.124–134, Janeiro, 2004.

[CLA 01] CLARKE, I. et al. Freenet: A distributed anonymous information storage and retrieval system. **Lecture Notes in Computer Science**, v.2009, p.46, 2001.

[CLA 02] CLARKE, I.; HONG, T. W.; SANDBERG, O. Protecting free expression online with Freenet. **IEEE Internet Computing**, v.6, n.1, p.40–49, Janeiro, 2002.

[COR 02] CORNELLI, F. et al. Choosing reputable servents in a P2P network. **In: Proceedings of the Eleventh International Conference on World Wide Web**, Honolulu, Hawaii, p.376 – 386, Maio, 2002.

[DAM 02] DAMIANI, E. et al. A reputation-based approach for choosing reliable resources in peer-to-peer networks. **In: Proceedings of the 9th ACM Conference on Computer and Communications (CCS'02)**, Washington, DC, USA, p.207 – 216, Novembro, 2002.

[dAM 03] DE ALMEIDA MATTOS, C. L. **Sentinel: Um Engenho Java para Controle de Acesso RBAC**. Trabalho final de graduação do Curso de Ciência da Computação na UFPB - Univesidade Federal Pernanbuco, 2003.

[DAS 03] DASWANI, N.; GARCIA-MOLINA, H. Open Problems in Data-Sharing Peer-to-Peer Systems. **In: Proceedings of 9th International Conference on Database Theory (ICDT 2003)**, Siena, Italy, p.1–15, Janeiro, 2003.

[DET 03] DETSH, A. Localização de Conteúdo em Redes Peer-to-Peer. **In: I Escola Regional de Redes de Computadores**, Porto Alegre, Rio Grande do Sul, Brasil, p.78–83, Setembro, 2003.

[EIK 04] EIKEMEIER, C.; LECHNER, U. Peer-to-Peer and Group Collaboration - Do they Always Match? **In: Proceedings of the 13th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises**, Linz, Austria, p.101–106, Junho, 2004.

[FAR 03] FARIAS, B.; SALGADO, A. C.; DO RÊGO GALVÃO, L. Conceptual Modeling of XML Schemas. **In: Proceedings of the 5th ACM International Workshop on Web Information and Data Management**, New Orleans, Lousiana, USA, p.102–105, Novembro, 2003.

[FEN 02] FENKAM, P.; DUSTDAR, S.; KIRDA, E. Towards an Access Control System for Mobile Peer-to-Peer Collaborative Environments. **In: Proceedings of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)**, Pittsburgh, PA, USA, p.95–100, Junho, 2002.

[FER 03] FERRAIOLO, D.; KUHN, R.; CHANDRAMOULI, R. **Role-Based Access Control**. Computer Security Series. Norwood, MA: Artech House, pp. 316, 2003.

[GAO 03] GAO, R. P2P Security and Trust. **Peer-to-Peer Journal (P2PJ)**, v.1, n.2, p.7–12, 2003.

[GM 03] GACIA-MOLINA, H.; YANG, B. PPay: micropayments for peer-to-peer systems. **In: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)**, Washington, DC, USA, p.300 – 310, Outubro, 2003.

[GOL 01] GOLLE, P. et al. Incentives for Sharing in Peer-to-Peer Networks. **Proceedings of the 3rd ACM Conference on Electronic Commerce**, Tampa, Florida, USA, p.264 – 267, Novembro, 2001.

[GUP 03] GUPTA, R. A Survey of Security Issues and Protocols in Peer-to-Peer Networks. Atlanta, US: CS 7210 Term Paper. College of Computing, Georgia Institute of Technology, Dezembro, 2003. Relatório técnico.

[HAL 02] HALEPOVIC, E.; DETERS, R. Building a P2P forum system with JXTA. **In: Proceedings of the Second International Conference on Peer-to-Peer Computing (P2P'02)**, Linköping, Sweden, p.41–48, Setembro, 2002.

[HAL 03] HALEPOVIC, E.; DATERS, R. The Costs of Using JXTA. **In: Proceedings of Third IEEE Internacional Conference on Peer-to-Peer Computing (P2P'03)**, Linköping, Sweden, p.160–168, Setembro, 2003.

[HAM 04] HAMADA, T. et al. Peer-to-Peer Traffic in Metro Networks: Analysis, Modeling and Policies. **In: IEEE/IFIP Network Operations and Management Symposium (NOMS'04)**, Seoul, Korea, p.425–438, Julho, 2004.

[HAR 68] HARDIN, G. The Tragedy of the Commons. **Science**, v.162, n.3859, p.1243–1248, Dezembro, 1968.

[IZA 04] IZAL, M. et al. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. **Lecture Notes in Computer Science – 5th International Workshop Passive and Active Network Measurement**, Antibes Juan-les-Pins, France, v.3015.

[KAR 03] KARAGIANNIS, T.; BOIDO, A.; FALOUTSOS, M. File-Sharing in the Internet: A Characterization of P2P Traffic in the Backbone. Department of Computer Science, Surge Building, Riverside, CA 92521.: University of California, Maio, 2003. Relatório técnico.

[KAR 04] KARAKAYA, M.; KORPEOGLUI, I.; ULUSOY, O. A Distributed and Measurement-based Framework Against Free Riding in Peer-to-Peer Networks. **In: Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing (P2P'04)**, Zurich, Switzerland., p.276–277, Agosto, 2004.

[KAW 04] KAWASHIMA, T.; MA, J. TOMSCOP – a synchronous P2P collaboration paltaform over JXTA. **In: Proceedings of 24th International Conference on Distributed Computing Systems**, Tokyo, Japan, p.85–90, Março, 2004.

[KOC 02] KOCH, M.; MANCINI, L.; PARISI-PRESICCE, F. A Graph-Based Formalism for RBAC. **ACM Transactions on Information and System Security**, v.5, n.3, p.332–365, Agosto, 2002.

[KOR 01] KORPELA, E. et al. SETI@home - Massively Distributed Computing for SETI. **IEEE Computing in Science & Engineering**, v.3, n.1, p.56–61, Fevereiro, 2001.

[KUR 04] KURMANOWYTSCH, R. **Omnix: An Open Peer-to-Peer Middleware Framework**. Argentinierstr, Austria: Vienna University of Technology, Fevereiro, 2004. Tese de Doutorado.

[LAN 01] LANDWEHR, C. E. Computer security. **International Journal of Information Security**, v.1, n.1, p.3–13, September, 2001.

[LEI 02] LEIBOWITZ, N. et al. Are File Swapping Networks Cacheable? Characterizing P2P Traffic. **In: Proceedings of the 7th International Worshop on Web Content Caching and Distribuition (WCW'02)**, Bouler, Colorado, USA, Agosto, 2002.

[LOO 03] LOO, A. W. The Future of Peer-to-Peer Computing. **Communications of ACM**, v.46, n.9, p.57–61, Setembro, 2003.

[LV 02] LV, Q. et al. Search and replication in unstructured peer-to-peer networks. **In: Proceedings of 16th International Conference on Supercomputing**, New York, USA, p.84–95, Junho, 2002.

[MAR 03] MARTI, S.; GARCIA-MOLINA, H. Identity Crisis: Anonymity vs. Reputation in P2P Systems. **In: Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03)**, Linköping, Sweden, p.134–141, Setembro, 2003.

[MIL 02] MILOJICIC, D. S. et al. Peer-to-Peer Computing. Hewlett-Packard Laboratories Palo Alto, Março, 2002. Relatório técnico.

[MIL 04] MILLER, J. Characterization of Data on the Gnutella Peer-to-Peer Network. **In: First Consumer Communications and Network Conference (CNCC'04)**, Las Vegas, Nevada, USA, p.498–494, Janeiro, 2004.

[MOT 03] MOTTA, G. H. M. B. **Um Modelo de Autorização Contextual para o Controle de Acesso ao Prontuário Eletrônico do Paciente em Ambientes Abertos e Distribuídos**. São Paulo, Brasil.: Escola Politécnica da Universidade de São Paulo, Julho, 2003. Tese de Doutorado.

[NEU 04] NEUBERG, B. Introduction to the Peer-to-Peer Sockets Project. New York, US: Columbia University, Maio, 2004. Relatório técnico.

[ORA 01] ORAM, A. **Peer-to-Peer: O poder Transformador das Redes Ponto a Ponto**, chapter Gnutella, p.105. Berkeley, São Paulo, SP, BR, Maio, 2001.

[P2P 04] P2P Architect Project. **Ensuring Dependability of P2P Applications at Architectural Level**. Site Web. Disponível em http://www.atc.gr/p2p_architect.

[PAP 04] PAPADIMITRIOU, C. et al. Free-Riding and Whitewashing in Peer-to-Peer Systems. **Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems**, Portland, Oregon, USA, p.228 – 236, Setembro, 2004.

[PAR 03] PARK, J. S.; HWANG, J. Role-based Access Control for Collaborative Enterprise In Peer-to-Peer Computing Environments. **In: Proceedings of 8th ACM Symposium on Access Control Models and Technologies (SACMAT 2003)**, Villa Gallia, Como, Italy, p.93–99, Junho, 2003.

[PER 95] PERNUL, G. Information Systems Security: Scope, State-of-the-art, and Evaluation of Techniques. **International Journal of Information Management**, v.15, n.3, p.239–255, 1995.

[RIG 04a] RIGHI, R.; PELLISSARI, F.; WESTPHALL, C. Escambo: Um Modelo de Comportamento e Reputação para Sistemas Peer-to-Peer. **In: II Escola Regional de Redes de Computadores (ERRC'04)**, Canoas, Rio Grande do Sul, Brasil, p.179–184, Julho, 2004.

[RIG 04b] RIGHI, R. et al. P2P-Limit: Uma Arquitetura para o Gerenciamento de Tráfego Peer-to-Peer em Backbones de Alta Velocidade. NPD-UFSC, Florianópolis, SC.: Univerisidade Federal de Santa Catarina / Ponto de Presença da RNP em Santa Catarina, Novembro, 2004. Relatório técnico.

[ROS 04] ROSSET, V. **Modelo de Arquitetura de Autorização e Distribuição de Direitos sobre Conteúdos Digitais**. UFSC: PPGCC – Universidade Federal de Santa Catarina, Fevereiro, 2004. Dissertação de Mestrado.

[ROU 04] ROUSSOPOULOS, M. et al. 2 P2P or Not 2 P2P? **In: Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS)**, San Diego, CA, USA, p.15–28, Fevereiro, 2004.

[SAD 03] SADOK, D. **Computação Colaborativa (P2P)**. Grupo de Trabalho da Rede Nacional de Pesquisa. Disponível em http://www.rnp.br/_arquivo/gt/2003/p2p.pdf.

[SAD 04] SADOK, D. et al. Peer-to-Peer: Computação Colaborativa na Internet. **In: Minicurso - Simpósio Brasileiro de Redes de Computadores (SBRC 2004)**, Gramado, Rio Grande do Sul, Brasil, p.3–43, Maio, 2004.

[SAN 94] SANDHU, R. S.; SARAMATI, P. Access Control: Principles and Practice. **IEEE Communications**, v.32, n.9, p.40–48, 1994.

[SAN 96] SANDHU, R. S. et al. Role-Based Access Control Models. **Computer Networks**, v.29, n.2, p.38–47, 1996.

[SCH 01] SCHOLLMEIER, R. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. **In: Proceedings of First IEEE International Conference on Peer-to-Peer Computing (P2P'01)**, Linköpings, Sweden, p.101–103, Agosto, 2001.

[SEN 04] SEN, S.; WANG, J. Analyzing Peer-to-Peer Traffic Across Large Networks. **IEEE/ACM Transactions on Networking**, v.12, n.2, p.219–232, Abril, 2004.

[SIN 03] SINGH, A.; LIU, L. TrustMe: Anonymous Management of Trust Relationships in Descentralized P2P. **In: Proceedings of the Third IEEE International Conference of Peer-to-Peer Computing (P2P'03)**, Lingköping, Sweden, p.142–149, Setembro, 2003.

[STA 03] STALLINGS, W. **Crytography and Network Security**, p.44. Prentice Hall, New Jersey, United States, 3th. ed., 2003. ISBN: 0-13-091429-0.

[STR 04] STRULO, B. Middleware to Motivate Co-operation in Peer-to-Peer Systems. **Peer-to-Peer Journal (P2PJ)**, v.1, n.5, p.1–12, Março, 2004.

[Sun 04] Sun Microsystems. **JXTA Technology: Creating Connected Communities**. Disponível em: http://www.jxta.org/docs/JXTA-Exec-Brief.pdf.

[TAL 03] TALIA, D.; TRUNFIO, P. Toward a Synergy Between P2P and Grids. **IEEE Internet Computing**, v.July/August 2003 issue, p.94–96, Agosto, 2003.

[TRI 04] TRIPUNITARA, M. Comparing the Expressive Power of Access Control Models. **In: Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04)**, Washington, DC, USA, p.62–71, Outubro, 2004.

[UCH 01] UCHOA, J. Q. **Políticas de Segurança e Políticas de Uso**. Palestra no Simpósio de Segurança em Informática (SSI 2001). ITA, São José dos Campos, Brasil.

[URU 04] URUSOV, D. JOSE – A Java Open Source Exchange. **Peer-to-Peer Journal**, v.1, n.4, p.12–17, Janeiro, 2004.

[VEN 03] VENTER, H.; ELOFF, J. A taxonomy for information security technologies. **Computers & Security**, v.22, n.4, p.299–307, Maio, 2003.

[VLA 04] VLACHOS, V.; ANDROUSSELLIS-THEOTOKIS, S.; SPINELLIS, D. Security applications of peer-to-peer networks. **Computer Networks: The International Journal of Computer and Telecommunications Networking**, v.45, n.2, p.109–205, Junho, 2004.

[WAL 02] WALLASH, D. S. A Survey of Peer-to-Peer Security Issues. **In: Proceedings of the International Symposium on Software Security**, Tokyo, Japan, Novembro, 2002.

[WAN 03] WANG, Y.; VASSILEVA, J. Trust and Reputation Model in Peer-to-Peer Networks. **In: Proceedings of the Third IEEE International Conference on Peer-to-Peer Computing (P2P'03)**, Linkoping, Sweden, p.150–159, Setembro, 2003.

[WIL 02] WILSON, B. J. **JXTA**. Berkeley, CA: New Riders Publishing, Junho, 2002. ISBN: 0735712344, 512 pp.

[YEA 02] YEAGER, W.; WILLIAMS, J. Secure Peer-to-Peer Networking: The JXTA Example. **IEEE IT Professional**, v.4, n.2, p.53–57, Março, 2002.

[ZHU 05] ZHU, Y.; HU, Y. Efficient, Proximity-Aware Load Balacing for DHT-Based P2P Systems. **IEEE Transactions on Parallel and Distributed Systems**, v.16, n.4, p.349–361, Abril, 2005.